



# POLE PASTORAL ZONES SECHES

CIRAD ■ CSE ■ ENEA ■ ISRA ■ UCAD

***Unité de Recherche en Partenariat***

**ADD 'Trans**

**WP 5**

**Bibliographie**

**Modélisation et Simulation Multi-agents:  
Concepts, Méthodes et Outils**

**Alassane Bah, Ibra Touré, Grégoire Leclerc**

**2006**

**Copyright PPZS**

## **Avant propos**

Ce premier travail de synthèse bibliographique sur les Systèmes Multi-Agents est né de la volonté de partager les différents concepts avec les partenaires du projet ADD Trans et particulièrement avec ceux des WP2 et WP5 qui partagent le même terrain d'étude au Sénégal.

# **LISTE DES FIGURES**

**Figure 1** : *Architecture agent Manta*

**Figure 2** : *Architecture Subsumption.*

**Figure 3** : *Architecture PRS*

**Figure 4** : *Architecture InteRRap*

**Figure 5** : *Représentation graphique d'un SMA (adapté de celui de Ferber)*

**Figure 6** : *Communication indirecte via l'environnement)*

**Figure 7** : *Communication directe entre acteurs*

**Figure 8** : *Syntaxe générale d'un message KQML*

**Figure 9** : *Syntaxe générale d'un message ACL*

**Figure 10** : *Graphe d'états du Protocole SIAN*

**Figure 11** : *Interface principale de CORMAS*

**Figure 12** : *diagramme de classes des entités spatiales, sociales et passives de la plate-forme Cormas*

**Figure 13** : *Espaces de simulation*

**Figure 14** : *Interface graphique pour visualiser les résultats de simulation sous CORMAS*

**Figure 15** : *Interface pour le paramétrage et le lancement de la simulation sous CORMAS.*

**Figure 16** : *représentation spatiale des différents éléments du simulateur.*

**Figure 17** : *Interface principale du simulateur « Pasteur »*

**Figure 18** : *Interface principale de la plateforme MIMOSA*

**Figure 19** : *Organisation du Modèle Aalaadin*

**Figure 20** : *Organisations concrètes*

**Figure 21** : *L'extension d'AGR avec environnement proposée par Parunak et Odell*

**Figure 22** : *Extension du Méta-modèle AGR*

**Figure 23** : *Les cinq calques de Cassiopée*

**Figure 24** : *Méta Modèle Gaia*

**Figure 25** : *Processus de conception GAIA*

**Figure 26** : *Processus mis en œuvre par ADELFE*

**Figure 27** : *Méta-modèle ADELFE*

**Figure 28** : *Différents modèles de CommonKads*

**Figure 29** : *Comparaison entre différentes méthodes agents*

## **LISTE DES TABLEAUX**

**Tableau 1** : *Classification des environnements de développement*

**Tableau 2** : *l'évolution des approches de programmation*

# Sommaire

<b>INTRODUCTION .....</b>	<b>6</b>
<b>I.     Rappel sur la simulation et la modélisation .....</b>	<b>7</b>
Modèles physiques-logiques-formels .....	7
Modèles statiques et dynamiques .....	7
Modèles linéaires et non linéaires .....	7
Modèles stables et instables .....	7
Modèles d'équilibre et de transition .....	7
<b>II.    Modélisation et simulation multi-agent .....</b>	<b>8</b>
Agents .....	8
Architectures d'agents .....	9
Systèmes multi-agents .....	12
Concepts : communications, coopération, coordination .....	13
La communication .....	13
Communication via l'environnement .....	13
Communication directe par envoi de message .....	13
Protocole d'interaction .....	16
Problème de Coordination .....	17
<b>III.   Outils et Plate-formes de simulation multi-agent .....</b>	<b>17</b>
CORMAS .....	19
1- la partie modélisation .....	19
2- la partie visualisation .....	21
SIMDELTA .....	22
SEALAB .....	22
PASTEUR .....	23
Le Contexte .....	23
Le simulateur (Pasteur) .....	24
Espace et ressources .....	24
Les agents .....	24
<b>IV.    Plateforme MIMOSA .....</b>	<b>25</b>
Méthodologies de conception orientées agents .....	26
Méthodologie orientée agents basé sur l'approche objet .....	27
Méthodologie AAI (Australian Artificial Intelligence Institute) .....	27
Méthodologie issue du projet Aalaadin .....	28
Méthodologie Cassiopée .....	30
Méthodologie GAIA .....	31
Méthodologie ADELFE .....	32
Méthodologie orientée agents basée sur l'ingénierie des connaissances et des besoins .....	34
Méthodologie MAS-CommonKADS .....	34
CommonKADS .....	34
Conclusion sur l'ingénierie orienté agent .....	36
<b>Conclusion .....</b>	<b>37</b>
<b>BIBLIOGRAPHIE .....</b>	<b>38</b>

## INTRODUCTION

Dans ce premier rapport, les principaux concepts SMA seront revisités et mis en perspectives. Différentes méthodes de conception de systèmes multi-agents seront également présentées. Il ne s'agira pas, dans cette partie, de discuter de la pertinence du choix de tel ou de tel autre type de modèle.

Ce travail doit nous permettre simplement de faire le point sur la démarche orientée agent qui est le type de modèle choisi pour étudier la dynamique des ressources naturelles et des paysages à l'échelle du territoire dans le cadre du WP 5 du projet « Add Trans ».

## I. Rappel sur la simulation et la modélisation

De manière générale, la **simulation** consiste non seulement en la reconstruction, à travers un modèle, d'un phénomène réel mais aussi en son utilisation (Shannon, 92). Elle est généralement perçue comme la construction d'un modèle du monde réel pour conduire des expériences (Gremy, 1990). Nous retiendrons la définition suivante de la simulation proposée par Hill dans (Hill, 1993) : « La simulation consiste à faire évoluer une abstraction d'un système au cours du temps afin d'aider à comprendre le fonctionnement et le comportement de ce système et à appréhender certaines de ses caractéristiques dynamiques dans l'objectif d'évaluer différentes décisions ».

Les notions de modèles et de simulation sont donc intimement liées. On parle souvent de modèle de simulation ou de modèle simulateur. La démarche généralement adoptée peut se décomposer en trois parties : - modélisation – expérimentation - validation.

La phase **modélisation**, destinée à la réalisation du monde artificiel, comprend la formulation d'hypothèses de travail, la collecte des données liées au problème puis la construction, à partir de ces données, du modèle proprement dit, en général fondé sur une théorie mathématique, logique, multi-agent ou autre. C'est cette théorie qui différencie les différents types de modèles et de simulations. Popper en donne une classification intéressante mais non exhaustive dans (Popper, 73).

### Modèles physiques-logiques-formels

Le modèle physique est une reproduction physique de la réalité (exemple : une maquette d'avion). Un modèle logique est une représentation simplement concevable de la réalité. La description y est purement verbale. Le modèle formel représente la réalité par des systèmes d'équations mathématiques (exemple : la loi de la chute des corps).

### Modèles statiques et dynamiques

Un modèle faisant intervenir la variable temps est qualifié de modèle dynamique. Dans le cas contraire, il est statique.

### Modèles linéaires et non linéaires

Dans un modèle linéaire les effets sont supposés proportionnels aux causes qui les ont provoqués. Dans un modèle non linéaire, il n'existe aucune proportionnalité entre une variable donnée et les variables qu'elle détermine.

### Modèles stables et instables

Dans un modèle physique l'état d'équilibre est dit stable si le corps peut récupérer sa position ou trajectoire d'origine, instable si l'état initial n'est pas récupéré après perturbation.

### Modèles d'équilibre et de transition

Un modèle qui cherche à représenter un phénomène conçu de manière générale et donc indépendamment du temps réel est un modèle d'équilibre.

Précisons, toutefois, pour qu'un modèle soit utilisable et performant, il faut qu'il présente un certain nombre de caractéristiques : la simplicité, la robustesse, la souplesse, l'adaptabilité, la totalité, la facilité d'accès. Une fois le modèle construit, il doit être utilisé. C'est l'étape de l'expérimentation. Cette phase consiste à faire varier certains paramètres du modèle afin de comprendre le phénomène à étudier. Le processus de **validation** permet de répondre à la question de la confiance que l'on peut accorder au modèle réalisé en confrontant les résultats obtenus lors de l'expérimentation et les observations du phénomène réel. La validation a, en d'autres termes, pour objet d'essayer de retrouver au niveau du modèle le mode de comportement relevé sur le système réel.

La simulation est donc un outil intéressant permettant de mieux comprendre et d'analyser des problèmes et phénomènes complexes. La modélisation et la simulation constituent à l'heure actuelle des outils performants et puissants. Jorgensen dans (Jorgensen, 1994) résume en quatre points les avantages de la modélisation :

- Les modèles ont leur utilité dans la surveillance de systèmes complexes.
- Les modèles peuvent être utilisés pour révéler les propriétés des systèmes écologiques.

- Les modèles peuvent montrer des carences dans nos connaissances et être utilisés pour définir des propriétés dans la recherche.
- Les modèles sont utiles pour tester des hypothèses scientifiques, dans la mesure où le modèle peut simuler les réactions de l'écosystème, lesquelles peuvent être comparées aux observations.

Ce sont ces différents éléments mis en exergue par Jorgensen qui justifient notre choix sur l'utilisation de la modélisation et la simulation pour étudier, tester des hypothèses, sur des problèmes de gestion de ressources naturelles et d'environnement. Les modèles représentent des outils de gestion et des instruments scientifiques. Dans cette synthèse nous nous intéressons particulièrement aux modèles de simulations orientées agents que nous proposons de décrire dans la section qui suit.

## II. Modélisation et simulation multi-agent

Les systèmes multi-agents (SMA) sont considérés comme une branche de l'intelligence artificielle distribuée (IAD). Les travaux de Hewitt sur la résolution de théorèmes et de Erman sur le tableau noir (blackboard) dans le cadre du projet HEARSAY II marquent le point de départ de cette discipline. Elle a bénéficié du développement de nombreux autres domaines comme la programmation orientée objet, la robotique, la programmation distribuée et bien d'autres.

Les systèmes multi-agents (SMA) connaissent actuellement un essor très important dans le domaine de la modélisation et de la simulation de systèmes complexes.

Ils sont de plus en plus utilisés non seulement pour le développement de logiciels (programmation orientée agent/interactions/organisation) et pour la résolution de problèmes (gestion de réseaux électroniques, ) mais aussi pour mieux comprendre, à travers la simulation, des éco-socio-systèmes qui combinent, à la fois, des dynamiques naturelles et des dynamiques sociales.

La simulation multi-agent apporte une vision novatrice au niveau modélisation et simulation dans les sciences de l'environnement, car elle utilise non seulement la représentation directe des individus « les agents », mais aussi celle de leurs interactions. Elle permet, ainsi, de résoudre des problèmes complexes associés à des sociétés dont le mode d'organisation au niveau macro résulte des différentes interactions au niveau micro, c'est-à-dire au niveau des agents, des individus. Elle offre en outre l'avantage à l'utilisateur d'être partie prenante de la simulation. Celui-ci utilise le simulateur « comme s'il s'agissait d'un laboratoire miniature, en déplaçant des individus, en changeant leur comportement et en modifiant les conditions environnementales » (Ferber 95).

La première étape d'une simulation multi-agent consiste à décomposer un phénomène réel généralement en un environnement auquel participe un ensemble fini d'objets autonomes et discrets associés par des liaisons statiques et dynamiques. La deuxième phase permet à partir de la définition de l'environnement (s'il existe), des objets et des liens qui les unissent de créer un modèle par transformation des objets en entités informatiques autonomes appelées agents. Quant à la dernière étape, commune à tous les types de simulation, elle consiste à raffiner le modèle en fonction des observations conjointes de la réalité et des simulations déduites du modèle.

### Agents

La définition de ce concept n'est pas tout à fait consensuelle. Il est polysémique et dépendant donc du domaine d'expertise considéré. Il est en effet associé à de nombreuses applications comme l'agent cognitif (Intelligence Artificielle), l'agent logiciel (démons Unix), l'agent assistant, etc. Les définitions sont nombreuses mais elles se rejoignent généralement sur certains points comme l'autonomie, l'action, la flexibilité. Jennings, Sycara et Wooldridge (Jennings et al. 98) proposent la définition suivante: " Un agent est un système informatique, situé dans un environnement, et qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu". Mais nous retiendrons la définition que nous propose Jacques Ferber dans son livre "Les systèmes multi-agents vers une intelligence collective" (Ferber, 95) comme référence. Ferber définit l'agent comme:

"une entité physique ou virtuelle



- qui est capable d'agir dans un environnement,
- qui peut communiquer directement avec d'autres agents,
- qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- qui possède des ressources propres,
- qui est capable de percevoir (mais de manière limitée) son environnement,
- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- qui possède des compétences et offre des services,
- qui peut éventuellement se reproduire,
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit."

## Architectures d'agents

Les caractéristiques que nous venons de présenter en terme d'autonomie, de communication, de représentation etc. ne s'appliquent toutefois pas à tout type d'agent. Selon le mode contrôle (concurrent / séquentiel) ou la forme de représentations des connaissances (symbolique/numérique), la description peut différer. Cette description peut être déclarative ou plutôt opérationnelle. Dans ce qui suit nous nous intéressons à l'aspect opérationnelle c'est-à-dire à l'architecture des agents.

Si nous nous plaçons dans une perspective de **Coordination** (*capacité de coordination des agents*), il est possible de classer les architectures d'agents en trois catégories :

### - Architecture d'agent autonome

Ces architectures exhibent des agents qui ont des capacités d'action et de perception de leur environnement. Par contre, leur faculté de coopération et de coordination sont minimales.

### - Architecture d'agent interagissant

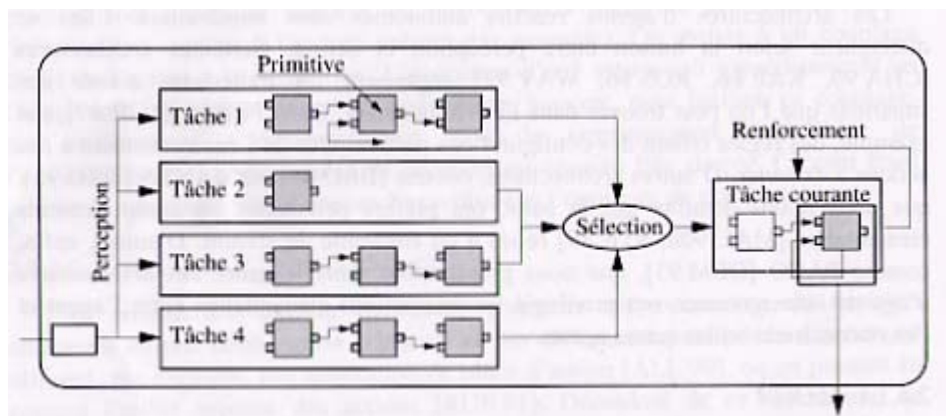
En plus des facultés d'action et de perception, ces agents ont une représentation des autres agents et peuvent interagir par l'envoi de messages.

### - Architecture d'agent social

L'architecture d'agent social possède des mécanismes de coordination et de coopération de haut niveau. D'un point de vue raisonnement, c'est-à-dire capacité de raisonnement des agents, nous avons le classement suivant :

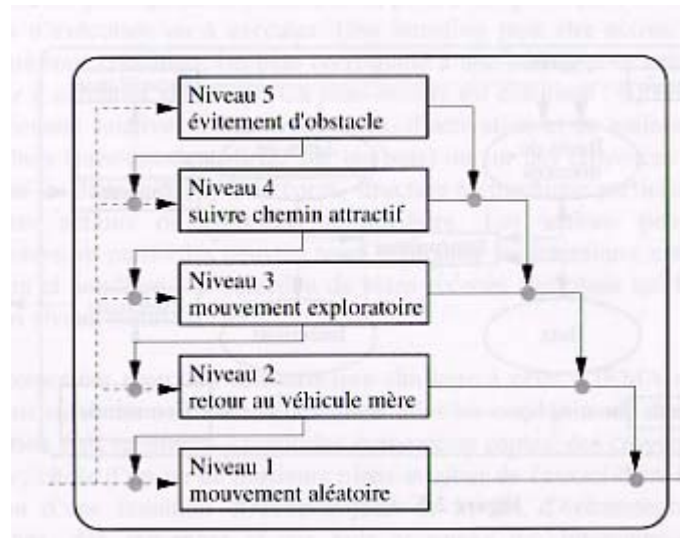
#### - Architecture d'agent réactif

Dans ce type d'architecture, les agents n'ont généralement aucune représentation symbolique de leur environnement. L'action suit immédiatement la manifestation du stimuli émis par l'environnement. De bons exemples pour illustrer ce type d'architecture sont le système Manta ( Modeling an ANThill Activity) développé par Drogoul (Drogoul, 1993) qui a été utilisé dans le domaine de la simulation éthologique (Figure 1) et le système développé par Brooks (Brooks, 1986) dans le domaine de la robotique qui repose sur ce qu'il appelle l'architecture « *Subsumption* » (Figure 2). Globalement les défenseurs de cette approche pensent que l'intelligence au niveau d'un SMA naît de l'interaction entre des **agents simples**.



**Figure 1 : Architecture agent Manta.**

L'architecture d'un agent MANTA comporte les opérations de **perception**, de **sélection** et d'**activation** qui manipulent un ensemble de tâches. Une tâche est une séquence d'actions primitives directement exécutables par les effecteurs de l'agent dans l'environnement. Elle est caractérisée par un **poids** (importance de la tâche au sein de l'agent), un **seuil** et un **niveau d'activation**. Après chaque exécution d'une action primitive de la tâche courante, l'opération de perception met en relation les stimuli issus de l'environnement avec les tâches correspondantes ; la **sélection** calcule le niveau d'activation de chacune des tâches en fonction de leur poids, de l'intensité des stimuli et choisit les tâches dont le niveau d'activation est supérieur à leur seuil. La tâche activée par l'opération d'activation sera celle dont le niveau d'activation est le plus élevé parmi les tâches choisies et la tâche courante : si la tâche courante est prolongée, son niveau d'activation est décrémenté, sinon le poids de la tâche activée est incrémenté et l'ancienne tâche est suspendue. On voit ainsi un mécanisme de sélection par compétition des tâches adapté en fonction de l'activité courante, ce qui permet d'éviter un comportement chaotique. Au niveau du système, cette régulation a notamment permis de mettre en évidence l'émergence de spécialisation et de répartition des tâches entre les agents (Boissier, 2001).

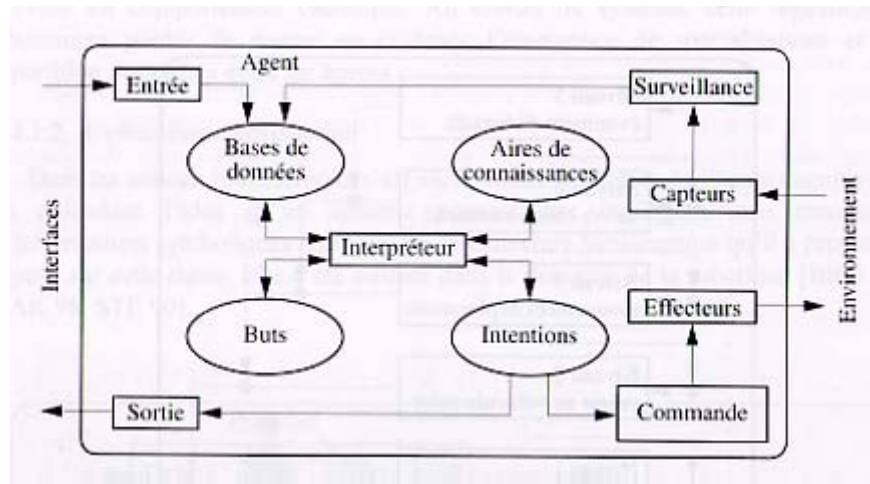


**Figure 2 : Architecture Subsumption.**

Cette architecture intercale entre la perception et l'action, différentes couches de comportement de complexité croissante : sur la couche basse, les comportements de base tels que le mouvement aléatoire ; sur les couches les plus hautes ceux devant permettre, théoriquement, de satisfaire des comportement spécifiques. Les couches fonctionnent en parallèle et accèdent à un sous-ensemble des informations captées. L'accès au système d'action est contrôlé via une hiérarchie de contrôle : les comportements présents sur une couche peuvent supplanter les comportements présents sur les couches inférieures en substituant leurs propres entrées aux entrées des couches inférieures et en inhibant toute sortie de ces mêmes couches pour des périodes de temps préprogrammées. Toute la difficulté de conception d'un tel agent réside dans la définition de la hiérarchie de contrôle entre les couches (Boissier, 2001).

#### - Architecture d'agent délibératif

De nombreuses architectures d'agent délibératif existent. Elles s'inspirent du modèle BDI (Belief Desire Intention) (Bratman, 1988). Ces agents sont généralement de structure très complexes contrairement aux agents réactifs. Un bon exemple est l'architecture PRS (Personal Reasoning System) (Georgeff 1987), (Georgeff, 1999) (Figure 3).

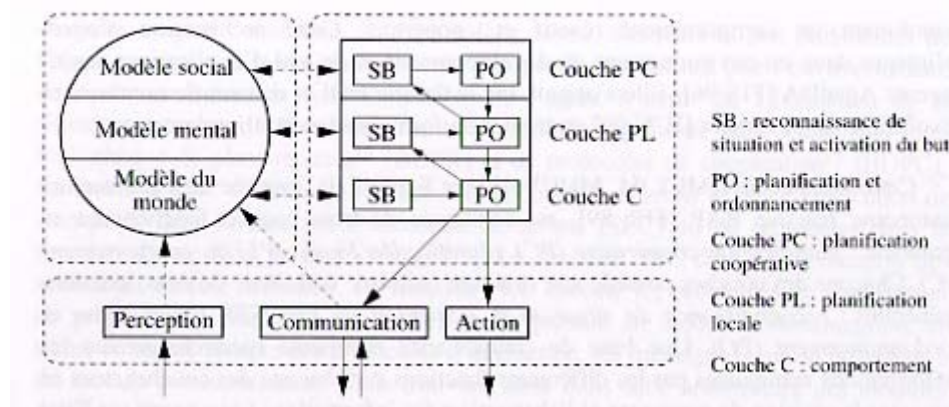


**Figure 3: Architecture PRS**

Cette architecture est constituée d'une bibliothèque de plans (Aires de connaissances), d'une base de croyances représentant des états du monde ou des déductions internes (Bases de données), d'une pile de buts (buts), d'une structure d'intentions (Intentions) et d'un ensemble d'opérations manipulant ces informations. Un but est la description du comportement désiré (interne ou externe) de l'agent exprimé en termes de satisfaction, de maintenance ou de vérification de propriétés. Les intentions correspondent à des files de priorité de plans partiellement instanciés, en cours d'exécution ou à exécuter. Une intention peut être active, suspendue ou suspendue sous condition. Un plan correspond à une recette pour satisfaire des buts ou réagir à certaines situations. Un plan-recette est constitué : (i) de conditions de déclenchement relatives à des événements, d'activation et de maintenance, portant sur des buts (raisonnement dirigé par les buts) ou sur des croyances (raisonnement dirigé par les données), (ii) d'un corps, structure hiérarchique partielle, décrivant les différentes actions ou sous-buts à satisfaire. Les actions peuvent modifier l'environnement mais elles peuvent aussi manipuler les intentions, croyances et buts de l'agent et conduire à la sélection des plans-recettes (les plans qui les contiennent sont d'un niveau méta) (Boissier, 2001).

#### - Architecture d'agent hybride

Les agents hybrides combinent généralement de structures simples et complexes. Ce type d'architecture a débuté dans le domaine de la robotique qui utilise un système réactif pour le contrôle de bas niveau et une planification pour la prise de décision. Nous citons comme exemple l'architecture InteRRaP (Figure 4) dont l'objectif est de proposer une gamme d'outils qui permettent de construire de systèmes complexes et dynamiques. Cette architecture a notamment été utilisée par Müller (Müller, 1994, 1996) dans le domaine de la robotique.



**Figure 4 : Architecture InteRRaP (Fis, 1995)**

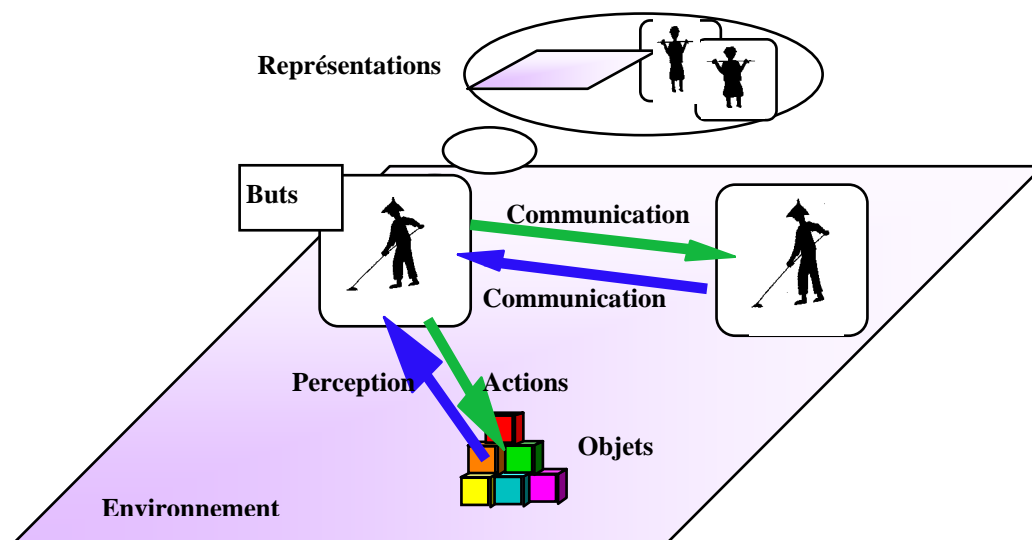
Cette architecture est constituée de trois couches fonctionnant en parallèle : Planification coopérative (PC), planification locale (PL) et comportement (C). Chacune des couches possède une structure uniforme constituée de deux fonctions générales : reconnaissance de situation et activation de but (SB), planification et ordonnancement (PO). Une base de connaissance représente hiérarchiquement les informations manipulées par les différentes fonctions sur chacune des couches, tout en assurant la révision de croyances et l'abstraction des informations : croyances sur l'état de l'environnement (modèle du monde) sur la couche C, croyances que l'agent a sur lui-même (modèle mental) sur la couche PL, croyances que l'agent peut avoir sur les autres agents du système (modèle social) sur la couche PC. Un composant constitué des fonctions de perception, de communication et d'action réalise l'interface avec l'environnement et les autres agents. La couche C réagit aux évolutions de l'environnement par l'utilisation des schémas de réaction (condition-action) déclenchés par des événements externes, et réalise des activités routinières (procédures de comportement). La couche PL contrôle la planification locale d'un agent en sélectionnant dans une bibliothèque de plans-recettes. Ceux-ci peuvent faire appel aux procédures de comportement de la couche C ou à d'autres sous-plans. La couche PC élargit les capacités de planification à un univers multi-agent en manipulant des plans communs comportant des points de synchronisation.[...]. Deux mécanismes de contrôle hiérarchique régulent les échanges entre les couches : requêtes ascendantes d'activation, signaux descendants d'engagements pour l'exécution. Une couche  $i$  ne devient active que si la couche inférieure ne peut pas traiter la situation courante : le processus  $Poi$  de la couche  $i$ , envoie une demande d'activation à  $Sb_{i+1}$  de la couche supérieure. Le résultat du traitement est renvoyé à  $Poi$ . Sur une couche  $i$ ,  $Poi$  exécute ses décisions en communiquant ses engagements à la couche inférieure  $Poi-1$ .

## Systèmes multi-agents

Un système multi-agent est un système distribué composé de plusieurs entités ou agents qui interagissent très souvent au sein d'un même environnement selon des modes de coopération, de concurrence ou de coexistence (Chaib-Draa et Levesque, 1996). Les SMA sont à l'intersection de plusieurs disciplines : programmation distribuée et génie logiciel, la robotique, la vie artificielle et autres. Ils s'inspirent aussi de plusieurs autres domaines connexes comme la sociologie, la psychologie, les sciences cognitives.

Selon Ferber (Ferber, 1995) un Système multi-agent est composé des éléments suivants (Figure 5) :

- Un environnement **E**, c'est-à-dire un espace disposant généralement d'une métrique.
- Un ensemble d'objets **O**. Ces objets sont situés, c'est-à-dire que, pour tout objet, il est possible, à un moment donné, d'associer une position dans **E**. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents
- Un ensemble **A** d'agents, qui sont les objets particuliers (**A** inclut **O**)
- Un ensemble de relations **R** qui unissent des objets (et donc des agents) entre eux.
- Un ensemble d'opérations **Op** permettant aux agents **A** de percevoir, produire, consommer, transformer et manipuler des objets de **O**.
- Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.



**Figure 5 : Représentation graphique d'un SMA (adapté de celui de Ferber)**

## Concepts : communications, coopération, coordination

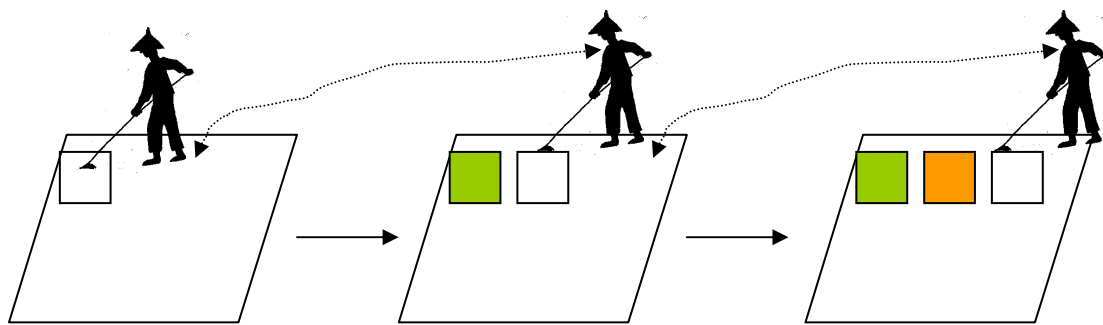
Le fait qu'il y ait plusieurs entités qui interagissent parfois au sein d'un même environnement implique nécessairement de se poser les questions liées entre autres aux problèmes de coordination, d'interactions (par communication ou autres) et de coopération.

### La communication

Nous distinguons principalement deux modes de communication (Koning, Pesty, 2001): une communication que l'on appelle indirecte par des signaux émis à travers l'environnement., C'est une communication directe entre agents qui consiste en l'envoi de messages (Figure 6).

#### Communication via l'environnement

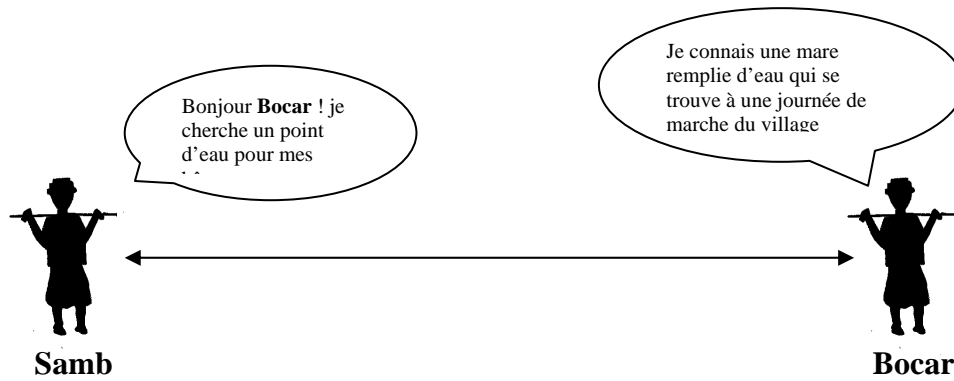
Ce type de communication se trouve généralement dans les modèles d'agents de type réactifs. Il s'agit simplement de propagation de stimuli ou de signaux via l'environnement. Ce type de communication est bien évidemment relativement restreint car le nombre de signaux est fini et leur interprétation fixe. Mais cela ne veut pas dire que ce type de modèles ne soit pas intéressant bien au contraire. La thèse de Alexis Drogoul (Drogoul, 1993) montre, s'il en était besoin, à travers le modèle Manta sur les fourmis, toute la pertinence de cette approche dite réactive par opposition à celle dite cognitive.



**Figure 6 : Communication indirecte via l'environnement.**

#### Communication directe par envoi de message

Ce mode de communication dit aussi intentionnel trouve son origine dans les travaux de Hewitt (Hewitt, 1977) sur les acteurs (le langage des acteurs). Cette approche de la communication concerne généralement les agents cognitifs. Dans ce cas de figure, l'agent est supposé posséder la capacité d'envoyer, de recevoir, d'interpréter et de répondre aux messages. Cet envoi peut se faire de manière synchrone ou asynchrone. Dans le cas synchrone, l'envoi de message n'est possible que si l'agent émetteur et récepteur conviennent d'un rendez-vous. La version asynchrone ne nécessite pas un rendez-vous entre agents car la *bufferisation* des messages est possible. Un exemple intéressant parmi d'autres pour illustrer ces deux modes de communication est le modèle « PASTEUR » (bah, 1997). Pendant les périodes de sécheresse les agents pasteur communiquent beaucoup pour trouver de meilleurs pâturages et des points d'eau pour l'abreuvement de leurs bêtes. Ces échanges d'information se font généralement de manière directe (Figure 7).



**Figure 7 : Communication directe entre acteurs**

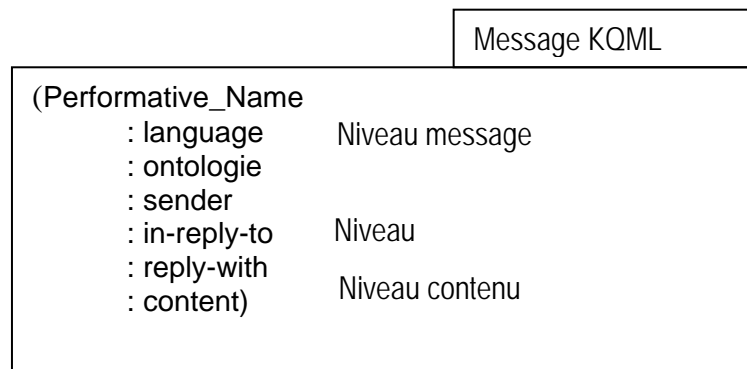
Rappelons, que la technique du tableau noir (blackboard) est très utilisée pour l'échange et le partage de connaissances. Au-delà du simple échange de messages et de connaissances, il s'est également développé grâce à la théorie des actes du langage fondée par Searle (Searle, 1969) sur la base des travaux d'Austin (Austin, 1962), tout un langage de communication entre agent dont les principaux sont : le langage KQML (Knowledge Query and Manipulation Language), et le langage ACL (Agent Communication Language).

#### - Le langage KQM

Le langage KQML a été proposé en 1993 par le consortium DARPA-KSE (Knowledge Sharing Effort) afin de développer des standards qui permettent le partage et la réutilisation de bases de connaissances et qui favorisent l'interopérabilité d'agents hétérogènes. Il a depuis lors connu de nombreuses évolutions notamment au niveau de la spécification de nouveaux performatifs. Un performatif est un ensemble d'actions que les agents doivent faire pour communiquer les uns avec les autres. Exemples : *ask-if*, *tell*, *deny*, etc.

Un message KQML se décompose en trois niveaux (Figure 8) :

- le niveau message qui permet d'identifier le type d'acte, le langage, l'ontologie utilisés qui s'appliquent au contenu ;
- le niveau communication qui permet d'identifier l'émetteur, le récepteur et le message ;
- le niveau contenu dont le format n'est pas imposé par le langage.



**Figure 8: Syntaxe générale d'un message KQML.**

#### - Le langage ACL (FIPA, 1999)



Le langage ACL développé par la FIPA<sup>1</sup> (Foundation for Intelligent Physical Agents) trouve ses bases, à l'instar du langage KQML, dans la théorie des actes de langage. Au-delà de certaines similarités notamment au niveau de la syntaxe des messages, un soin particulier a été apporté par les concepteurs au niveau de la sémantique et de la description formelle des actes de communication. Ils proposent, à cet effet, un langage spécifique appelé **Semantic Language** (SL) pour décrire les différents actes de communication. Ce langage s'appuie sur la logique modale du premier ordre (Sadek, 1991).

Sans entrer dans les détails, nous rappelons que le modèle mental d'un agent est décrit par trois attitudes (Figure 9) :

- **Belief** formalisé par l'opérateur modal **B**
- **Uncertainty** formalisé par l'opérateur modal **U**
- **Choice** formalisé par l'opérateur modal **C**.

**B<sub>i</sub> P** : l'agent <sub>i</sub> croit (implicitement) que la proposition **P** est vraie ;

**U<sub>i</sub> P** : l'agent <sub>i</sub> est incertain à propos de **P** mais pense que **P** est plus vraisemblable que  $\neg P$

**U<sub>i</sub> P** : l'agent <sub>i</sub> souhaite que la proposition **P** soit vérifiée.

A ces opérateurs modaux s'ajoutent d'autres opérateurs qui cette fois-ci permettent de formaliser l'action (**a<sub>i</sub>**). Ces opérateurs sont au nombre de trois : **Feasable**, **Done** et **Agent**.

**Feasable (a, p)** : l'action **a** peut être effectuée et si tel est le cas la proposition **p** sera alors vraie ;

**Done (a, p)** : l'action **a** a été effectuée et la proposition **p** est vraie depuis ;

**Agent (i, a)** : l'agent **i** est le seul effectuant ou allant effectuer l'action **a**.

A cela s'ajoute deux autres concepts :

- **Persistent goal** noté **PG**, **PG<sub>i</sub> p** signifie que l'agent <sub>i</sub> a **p** comme but persistant ;
- Et **Intention** noté **I**, **I<sub>i</sub> p** signifie que l'agent <sub>i</sub> à l'intention de réaliser **p**.

ACL possède quatre actes de langage dits primitifs (**inform**, **request**, **confirm**, **disconfirm**) et dix huit actes composés (**query\_if**, **request\_when**, **reject\_proposal**, etc.). Le format général de tout acte est le suivant :

<**i, act (j,p)**> avec

- **i**, représente l'agent émetteur ;
- **act**, l'acte de communication ;
- **j**, l'agent receveur ;
- **p**, la proposition.

A chaque acte est aussi associé :

- des pré-conditions notées **FP** (**Feasability Preconditions**) qui doivent être vraies pour l'acte puisse être utilisé ;
- les effets attendus après utilisation de l'acte notés **RE** (**Rational Effects**)

Nous pouvons ainsi traduire l'acte de langage suivant :

« L'agent **i** demande à l'agent **j** de lui dire s'il pleut »

<sup>1</sup>Il s'agit d'une organisation internationale à but non lucratif, créée en 1996, qui s'occupe de la standardisation des technologies agents. <http://www.fipa.org>

$\Phi$  : « il pleut »

**Query\_if** :  $\langle i, \text{query-if} (j, \Phi) \rangle \equiv \langle i, \text{request} (j, \langle \text{inform-if} (i, \Phi) \rangle) \rangle$

**FP** :  $\neg \text{Bif}_i \Phi \wedge \neg \text{Uif}_i \Phi \wedge \neg \text{B}_i \text{I}_j \text{Done} (\langle j, \text{inform-if} (i, \Phi) \rangle)$

**RE** :  $\text{Done} (\langle j, \text{inform} (i, \Phi) \rangle \mid \langle j, \text{inform} (i, \neg \Phi) \rangle)$

L'agent  $i$  demande à l'agent  $j$  qu'il l'informe si  $\Phi$  est vrai ou non

**Inform\_if** :  $\langle i, \text{inform-if} (j, \Phi) \rangle \equiv \langle i, \text{inform} (j, \Phi) \rangle \mid \langle i, \text{inform} (j, \neg \Phi) \rangle$

**FP** :  $\text{Bif}_i \Phi \wedge \neg \text{B}_i (\text{Bif}_j \Phi \vee \text{Uif}_j \Phi)$

**RE** :  $\text{Bif}_j \Phi$

Avec **Bif**  $_i \Phi \equiv \text{B}_i \Phi \vee \text{B}_i \neg \Phi$

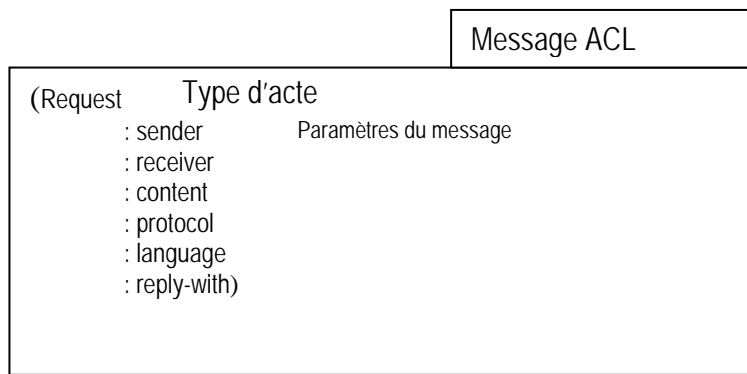


Figure 9: Syntaxe générale d'un message ACL

## Protocole d'interaction

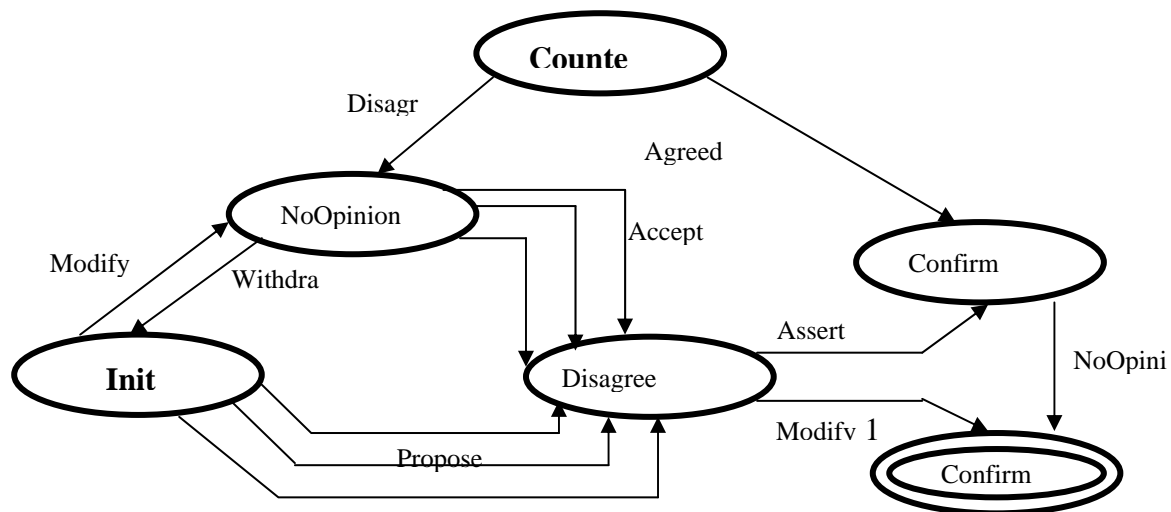
Pour que la communication, c'est-à-dire l'échange de messages fonctionne bien, il est nécessaire de définir des protocoles qui permettent de guider les interactions entre les différents agents. Un protocole précise qui peut dire quoi à qui et les réactions possibles à ce qui est dit par l'autre partie (Koning et Pesty, 2001). Plusieurs protocoles existent : le protocole d'apprentissage coopératif (Sian, 91) (fig.), le protocole de vente aux enchères (Rodrigues-Aguilar, 1999), etc. Mais, le plus connu est le *Contract Net Protocol* (CNP) (Smith, 1980). L'intérêt de celui-ci réside dans le fait qu'il permet aux agents :

- d'interagir pour résoudre un problème global ;
- de partager dynamiquement la charge de travail ;
- et d'intégrer leurs résultats.

Ce protocole facilite ainsi, la coopération entre agents en leur permettant de demander de l'aide lorsqu'ils sont confrontés à un problème particulier qu'ils n'arrivent pas à résoudre (Figure 10).

Plusieurs manières de représenter de manière expressive ces protocoles existent dans la littérature. Parmi celles-ci il y a : les graphes d'état-transition (Figure 25), le langage formel **Z** basé sur la théorie des ensembles et la logique du premier ordre (Spivey, 1992), les réseaux de Petri, le langage AgentTalk (Kuwabara, 1995) et bien d'autres.





**Figure 10: Graphe d'états du Protocole SIAN (Sian, 1991).**

## Problème de Coordination

La problématique de la coordination est importante car elle s'intéresse à non seulement comment «aligner des activités de différentes perspectives » (Bond et al. 1988) c'est à dire faire interagir des agents au point de vue conflictuels, mais aussi à comment assurer la cohérence globale du système multi-agent. Katia Sycara (Sycara, 1991) pense que « la coordination peut être définie comme la gestion des interdépendances entre les activités des agents ou, de manière plus pragmatique, comme le choix, l'ordonnancement et l'affectation des actions au cours du temps pour essayer de maximiser l'ensemble des critères de décision qui peut varier dans le temps ». Cela revient donc à se poser et à répondre aux questions suivantes:

- que coordonner ;
- à travers quelle représentation ;
- comment ?

Une catégorisation simplifiée mais intéressante des modes de coordination nous est proposée par Sascha Ossowski (Ossowski, 1999) qui distingue :

- la coordination centralisée qui suppose l'existence d'un coordinateur central ;
- de la coordination distribuée dont l'objet est de trouver un compris entre des agents qui ont généralement des intérêts divergents. Ce dernier mode présuppose un mécanisme de négociation (Müller, 1996).

On distingue, par ailleurs, deux modèles de mécanisme de coordination : les modèles orientés tâches, et les modèles orientés agents. Ces mécanismes donnent une réponse au « comment coordonner ? ». Les modèles orientés tâches s'inscrivent dans un contexte de résolution distribuée de problèmes ou l'on a un but global que doivent satisfaire les agents en exécutant chacun une sous-tâche bien précise (Lesser, 1981) (Durfee, 1991). On se retrouve de fait dans une situation de coopération. Les modèles orientés agents ne partagent pas nécessairement un but global commun. Chacun cherche à maximiser sa satisfaction individuelle.

En réalité, les problèmes de coordination, de coopération, de négociation et plus généralement d'interaction sont très étroitement liés. Non seulement coordonner, coopérer, négocier c'est interagir mais aussi pour coordonner il faut généralement faire négocier et peut-être faire coopérer plusieurs agents.

## III. Outils et Plate-formes de simulation multi-agent

Durant cette dernière décennie, nous avons constaté un foisonnement d'outils, de plates-formes ou plus généralement d'environnements de développement pour systèmes multi-agents. Ces environnements ont permis des avancées significatives dans les recherches en permettant à des utilisateurs d'horizons très divers (pédologue, écologue, géographes, informaticiens et autres) de

concevoir avec une certaine facilité leurs applications sans perdre trop de temps à comprendre les différents concepts. Nous classons ces environnements en cinq catégories (Gessoum et Ocelllo, 2001) (Tableau 1):

Catégorie	Environnement
Outils pour la simulation	Swarm (Minar, 1996), Cormas (Bousquet et al., 1998), Geamas (Marcenac, 97), Mice (Durfée, 1988)
Outils pour l'implémentation	Actalk (Briot, 1989), DECAF (Graham, 2000), Magique (Bensaid 1997), Mocah (Abchiche, 1999)
Outils pour la conception	Alaadin (Ferber, 1998), Maleva (Lhuillier, 1998), JAF(Vincent, 2001)
Outils pour la conception et l'implémentation	AgentBuilder (AGE, 1999), DIMA (Guessoum, 1997), MACE(Gasser, 1987),
Outils pour la conception, l'implémentation et la validation	Desire (Brazier, 1995), MAST (Hannoun, 1999), Zeus (Nwama, 1999)

*Tableau 1: Classification des environnements de développement (d'après Guessoum et Ocelllo, 2001)*

*- environnement pour la simulation*

Ce type d'outil fournit un ensemble de bibliothèques qui facilite le développement de simulation agents. Ils intègrent généralement un noyau de simulation à événements discrets qui permet d'observer, de comprendre et de contrôler les différents agents. Un bon exemple en est la plate-forme Cormas (Bousquet et al., 1998) que nous présenterons de manière détaillée par la suite.

*- environnement pour l'implémentation*

Ces outils implémentent des architectures d'agents qui vont d'agents simple mais actifs aux agents dotés d'un langage de communication plus ou moins complexe : KQML, ACL.

*- environnement pour la conception*

Les environnements pour la conception sont généralement basés sur un modèle de type componentiel. Le système multi-agent est conçu comme étant un ensemble organisé de composants logiciels. Un agent dans ce système est une association de composants simples ou complexes (Maleva).

*- environnement pour la conception et l'implémentation*

Ces environnements offrent un ensemble d'outils qui permettent de définir de groupes d'agents. Ils fournissent également un ensemble de facilités pour le développement indépendamment de l'architecture d'agent choisie.

*- environnement pour la conception, l'implémentation et la validation*

En plus de la conception et de l'implémentation, ces environnements offrent des outils de validation pour les applications multi-agents développés (MAST).

Cette richesse ne doit cependant pas cacher un réel défi de généricité, de facilité d'utilisation, d'autant plus que les problèmes abordés sont souvent très complexes : échelles spatio-temporelles différentes, richesse et hétérogénéité des agents en jeu.

Nous dirons à la lumière de ce tour d'horizon qui est loin d'être exhaustif qu'un bon environnement doit :

- être ouvert ;
- doit intégrer l'hétérogénéité ;
- doit fournir des outils de suivi ou de simulation ;
- doit permettre la validation ;

- doit offrir des capacités de déploiement pour répondre aux besoins d'exécution distribuée et de sécurité.

En plus de ces plate-formes qui se veulent générique, il existe un très grand nombre d'outils ou d'environnements que nous appelons dédiés. C'est-à-dire qui ne sont développés que pour étudier un problème particulier. Nous citons en exemple :

- SHADOC, sur les périmètres irrigués au Sénégal avec Olivier Barreteau (Barreteau, 1998),
- SIMDELTA (Bousquet et al., 1993), sur la pêche au niveau du delta central du fleuve Niger au Mali.
- SEALAB (Le Page, 96) sur la reproduction des poissons.
- PASTEUR (Bah et al., 98) sur la mobilité pastorale en zone sahélienne.

Nous présentons, de manière un peu plus détaillée, dans ce qui suit la plate-forme CORMAS ainsi que les simulateurs SIMDELTA, SEALAB et PASTEUR. Nous terminerons par la présentation de la plateforme MIMOSA en cours de production sous la houlette de Jean-Pierre Müller (Müller, 2003).

## CORMAS

CORMAS (Bousquet et al. 98) est un environnement de programmation destiné à l'élaboration de modèles multi-agents à des fins de simulation. Il s'agit d'une plate-forme très ouverte avec laquelle on peut réaliser des modèles d'une très grande hétérogénéité. Cependant, elle doit son succès à la facilité qu'elle offre pour la réalisation de modèles qui s'intéressent à la problématique de la gestion des ressources naturelles renouvelables. Un grand nombre de modèles divers et variés utilisant cette plate-forme existent. Pour de plus amples informations nous suggérons la visite du site dédié à CORMAS : <http://cormas.cirad.fr>.

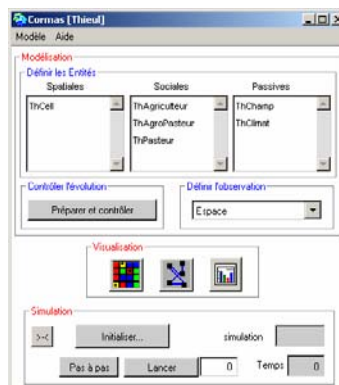
Cormas a été développée à partir de l'environnement VisualWorks dont le langage de programmation est le Smalltalk. Au niveau de l'interface principale (Figure 11), on distingue trois grandes parties :

### 1- la partie modélisation

Cette partie permet à l'utilisateur de définir non seulement les différentes entités de son modèle mais aussi de spécifier les modes d'interactions entre celles-ci. Dans le cas des entités sociales par exemple, il est possible d'implémenter des procédures de communication directe (envoi de message) ou indirecte (via l'environnement). Cormas distingue trois grandes classes d'entités (Figure 12) : diagramme de classes):

- les entités sociales,
- les entités spatiales
- les entités passives.

Cette décomposition en trois classes d'entités génériques (sociales, spatiales, passives) montre la spécificité de la plate-forme qui s'intéresse principalement à la problématique de la gestion des ressources renouvelables c'est-à-dire aux interactions entre dynamiques sociales et environnementales.

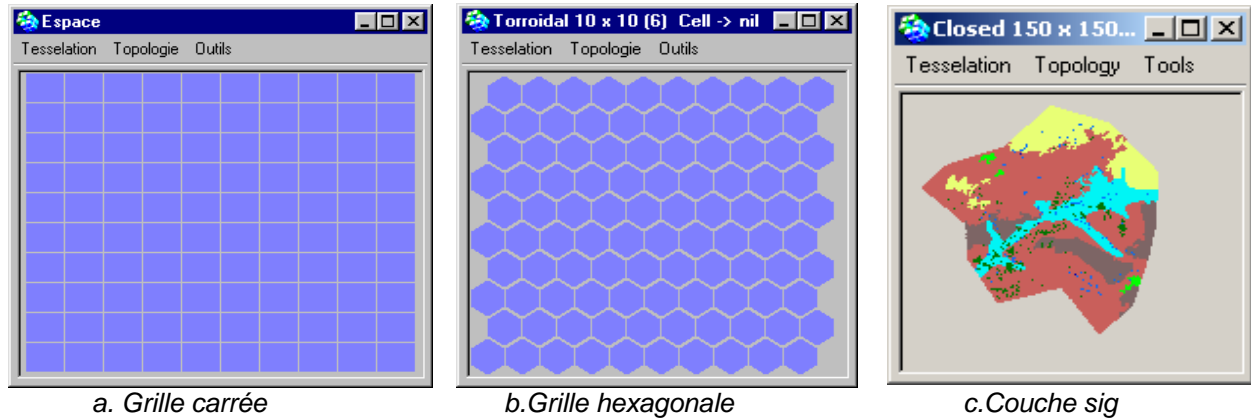


**Figure 11 : Interface principale de CORMAS**



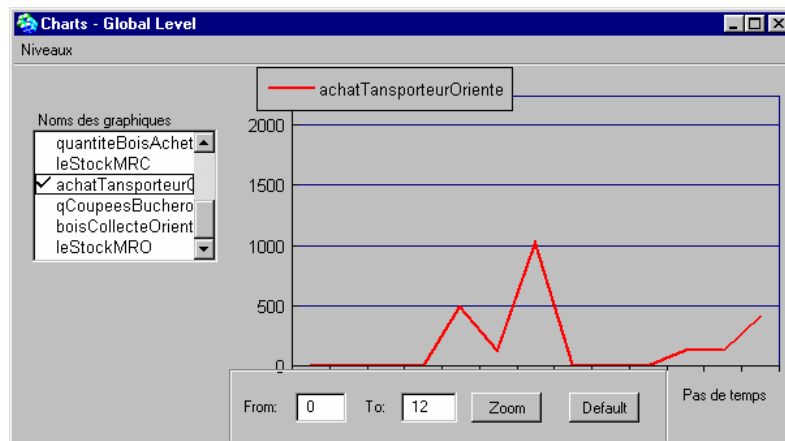
## 2- la partie visualisation

Cette partie nous offre la possibilité de suivre à travers une interface appelée « espace de simulation » l'évolution des entités sociales et la dynamique des entités spatiales (Figure 13). Cet espace est souvent une grille de dimension variable dont les cellules peuvent être de forme hexagonales, carrées ou rectangulaires (a,b). Cet espace peut par ailleurs être couplé avec un Système d'Information Géographique (SIG).et importé des couches vecteur ou raster (c).



**Figure 13: Espaces de simulation**

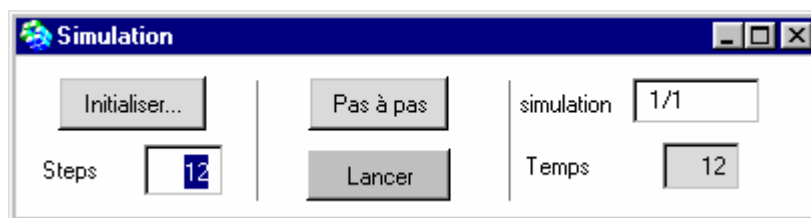
Il est possible également non seulement de visualiser les communications entre les agents mais aussi d'obtenir différents graphiques (sous Excel ou autre) à l'issue des simulations. (Figure 14)



**Figure 14 : Interface graphique pour visualiser les résultats de simulation sous CORMAS**

### 3- la partie simulation

La partie simulation permet au modélisateur de « faire évoluer » son monde virtuel dans le temps et dans l'espace en choisissant un pas de temps qui lui convient (Figure 15).



*Figure 15 : Interface pour le paramétrage et le lancement de la simulation sous CORMAS.*

## SIMDELTA

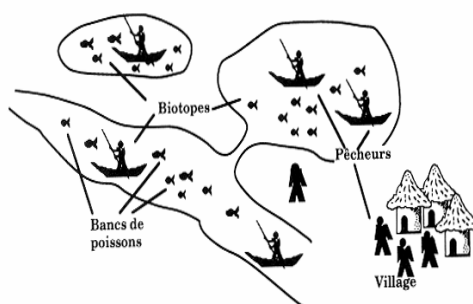
Le simulateur SIMDELTA (Bousquet et al., 1993), développé à partir du langage à objets SMALLTALK, à été réalisé par F. Bousquet et C. Cambier dans le but de synthétiser les connaissances sur la pêche au niveau du delta central du fleuve Niger au Mali. Le système étudié est constitué de différents acteurs (pêcheurs, poissons, biotopes) qui interagissent à différents niveaux de complexité mais aussi de chercheurs qui donnent leur vision de ce monde à travers des connaissances, des modèles et des théories (Figure 16). C'est à la fois des interactions entre acteurs et des interactions entre des connaissances qui sont représentées. SIMDELTA est utilisé pour mener diverses expériences :

- sur la dynamique des populations de poissons ;
- sur la prise de décisions des pêcheurs.

Entre autres résultats, on observe l'évolution des captures qui suit une courbe en trois phases :

- une augmentation des prises avec l'effort ;
- une phase plateau où les captures restent stables malgré l'augmentation de l'effort ;
- chute des captures.

Pour en savoir plus nous proposons de lire la thèse de François Bousquet (Bousquet, 94) intitulée : « Des Milieux , Des Poissons, Des Hommes : Etude par simulations multi-agents ».



*Figure 16 : représentation spatiale des différents éléments du simulateur.*

## SEALAB

SEALAB (Le Page, 96) a été développé afin non seulement d'étudier l'influence d'un milieu hétérogène et fluctuant sur la recherche, par des individus (poissons), d'un site reproductif mais aussi d'évaluer

leurs conséquences sur la dynamique globale de la population de poissons. L'environnement artificiel de SEALAB (Figure 17) se présente sous la forme d'une grille de cellules spatiales où les poissons se déplacent de cellules en cellules jusqu'au moment où ils décident de se reproduire. Chaque cellule est caractérisée par un indice synthétique qui traduit l'influence de l'ensemble des facteurs hydro-climatiques sur le processus reproductif.

Deux hypothèses théoriques de comportement des recherches sont comparées. La première postule qu'un individu s'obstine à rechercher les mêmes conditions environnementales que celles qui prévalaient lors de sa propre naissance. La seconde suppose qu'un individu recherche les conditions environnementales « optimales » (c'est-à-dire procurant le plus grand succès reproductif). Une des grandes conclusions de ce travail est qu'une population mixte résiste mieux en cas de survenue d'une anomalie hydro-climatique.

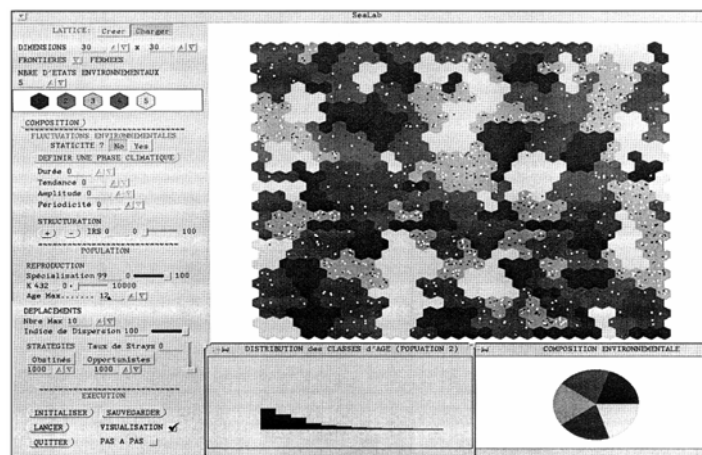


Figure 17: Le simulateur SEALAB.

## PASTEUR

Nous détaillons la présentation de cette plateforme car elle est à notre connaissance la première à être développée sur le thème du pastoralisme en général au Sénégal.

### Le Contexte

Au Sahel, la question des interactions entre pratiques pastorales et environnement est au centre des débats sur le développement durable de la région (Milleville 1989 et 1992, Bernus 1990, Behnke et Scoones 1992, Claude *et al.* 1991, Boutrais 1992, Blanc-Pamard et Boutrais 1994, Collectif 1995, Thebaut 1990 et 1995). Les principaux thèmes de discussion concernent la *mobilité* des troupeaux sahéliens, nécessaire pour les uns, dépassée pour les autres, l'impact environnemental de l'élevage pastoral et les influences des politiques socio-économiques sur les comportements des éleveurs. En effet, les politiques dirigistes de développement et de sécurisation de l'élevage pastoral se heurtent à des systèmes très mobiles, adaptés à la variabilité temporelle et spatiale du milieu, sur lesquels diagnostics et identification d'appuis sont difficiles. Ainsi jusqu'à présent, l'intensification de l'élevage sahélien a été un échec face aux contraintes économiques et climatiques de la région (Benoit 1984, Thebaut 1995). Les hécatombes connues dans les années 80 autour de certains forages bien alimentés en eau ont souligné les difficultés d'une artificialisation toujours, quoiqu'on fasse, tributaire du milieu.

La nécessité d'une modélisation du "phénomène" s'impose alors. Plutôt que de continuer à expérimenter en grandeur réelle des politiques dont les conséquences s'avèrent ensuite dramatiques au niveau humain ou environnemental, la simulation informatique des interactions élevage-environnement, antérieures puis issues de ces politiques, serait à la fois plus prudente et plus rigoureuse.

Cependant, devant les avis très divergents des experts sur la mobilité pastorale, les rationalités économiques et sociales des éleveurs, la valeur adaptative des pratiques pastorales face à un environnement changeant et diversifié, l'idéal serait d'aider le pastoraliste à évacuer au maximum, dans l'expérimentation modélisatrice à mettre en place, les hypothèses initiales sur ces trois questions. En effet, les fortes divergences d'opinions des pastoralistes et les nombreux échecs des opérations de développement soulignent deux traits essentiels du pastoralisme sahélien : (i) la complexité d'une situation qui allie des usagers combinant plusieurs cycles de mobilité (quotidienne, saisonnière et pluriannuelle) (Milleville 1992, d'Aquino 1996) à des ressources aléatoires dans l'espace et le temps ; (ii) la faiblesse des connaissances sur la nature, la qualité et l'évolution des interactions entre les comportements des usagers et leur environnement.

### Le simulateur (Pasteur)

Ainsi, afin d'évaluer les meilleures pratiques pastorales et les politiques de développement les plus utiles à l'échelle d'une région sahélienne, un système multi-agents a été conçu. Il utilise l'environnement Cormas (Bousquet et al., 1998), et peut donc se décomposer d'une part en une modélisation de l'espace et des ressources et d'autre part en une modélisation des comportements des agents.

### Espace et ressources

L'environnement pastoral est représenté par une grille où sont dispersées, selon une proportion modifiable à volonté par l'utilisateur, cinq ressources spécifiques : quatre *sensu stricto*, l'eau, le pâturage "moyen" et deux fourrages plus rares, appelées ressources A et B (qui peuvent représenter des ressources protéiques particulières ou une cure salée) et une "ressource sociale", le village. Certaines ressources se renouvellent périodiquement et le taux de renouvellement est associé au climat.

Il est très facile pour l'utilisateur de rajouter une autre "ressource sociale" si le besoin s'en fait sentir au cours de l'expérimentation, voire d'autres ressources biologiques. Le choix a été fait de ne pas se focaliser sur une implémentation plus fine de l'environnement biologique (cycle d'évolution des différents types de végétation, résistances différentes à la pâture et à la sécheresse,...), étant donné le niveau pris en compte par la simulation, que ce soit spatial (espace régional) ou social (règles collectives d'usage et programme de développement).

### Les agents

Les agents représentent le couple troupeau-pasteur : le comportement de l'agent troupeau est caractérisé par une consommation des ressources et des déplacements. Comme dans la plupart des systèmes multi-agents nous avons doté l'agent troupeau des capacités de représentation, mémorisation, et communication. La perception de l'expert est reprise (testée) dans :

- L'évaluation qualitative des besoins d'un troupeau ; à chaque ressource, sont associées deux valeurs. La première, variable, dite niveau, représente l'état de satisfaction de l'agent par rapport à la ressource. Elle décroît dans le cas où l'agent ne disposerait pas régulièrement de cette ressource. La seconde, fixe, est dite seuil critique. Si le niveau correspondant à une ressource atteint le seuil critique, le besoin pour cette ressource devient urgent et l'agent doit agir en conséquence.
- Les différentes logiques possibles de déplacement du troupeau pour satisfaire ses besoins. Selon quelle rationalité le troupeau consomme-t-il (ou doit-il consommer) les différentes ressources disponibles sur le territoire ? Les agents sont capables ou non de mémoriser la position des ressources. Deux modes de mémorisation sont proposés, local ou global, selon le degré de perception de l'agent. Les agents peuvent être dotés ou non de la faculté de communiquer. Dans le cas où ils communiqueraient et disposeraient d'une mémoire, ils peuvent s'échanger le contenu partiel ou total de cette mémoire. S'ils ne disposent pas de mémoire mais peuvent cependant communiquer, ils n'échangeront que la position des ressources qu'ils perçoivent au moment de la communication.



Les critères d'analyse des résultats de simulations, choisis par le pastoraliste, sont les suivants (sur un cycle de un à dix ans) : survie des troupeaux sans et avec sécheresses, degré de mobilité et de territorialisation apparaissant, état et résilience de l'environnement.

En résumé, pour une telle problématique pastorale caractérisée par la complexité des interactions à prendre en compte (variabilité extrême des ressources dans l'espace et le temps, à laquelle répondent des logiques de mobilité des usagers se référant à plusieurs échelles de gestion, du village à la sous-région (d'Aquino 1996)), le choix a ici été fait de s'appuyer sur très peu d'*a priori* dans le modèle multi-agents. Le problème de la mobilité pastorale et du degré de territorialisation ne sont pas déterminés par l'expert : ils apparaîtront, ou non, selon les logiques de déplacement retenues pour les éleveurs, l'évolution de l'environnement induit par ces logiques et les conditions climatiques, ainsi que les conditions imposées par les politiques de développement testées. Quelle peut être ensuite la logique de l'aménagement du territoire et des règles collectives ?

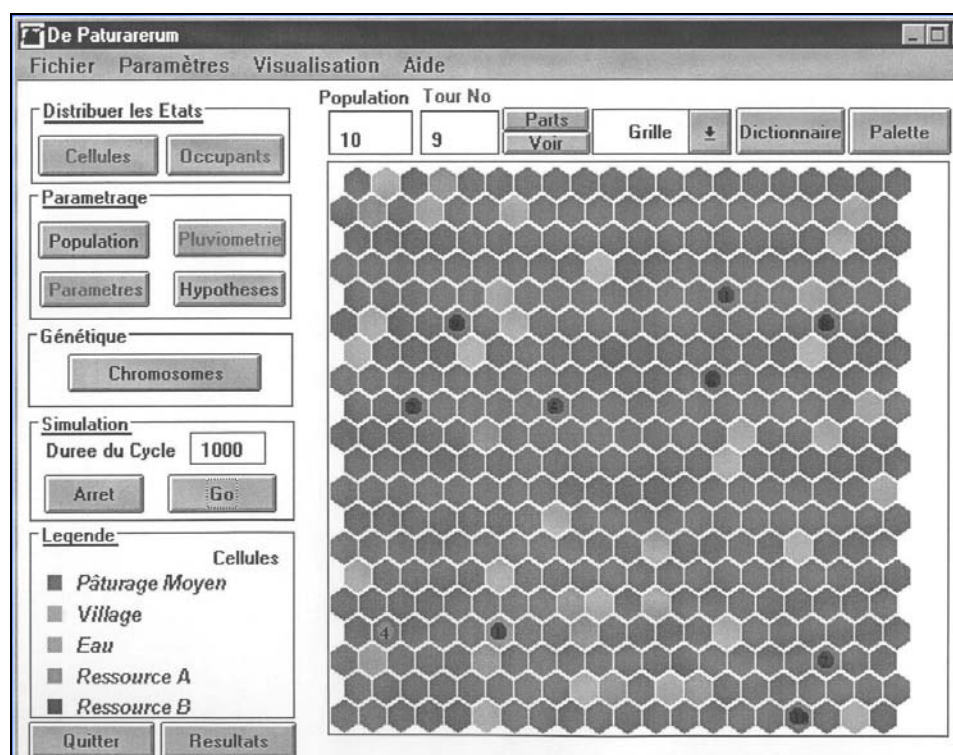


Figure 17: Interface principale du simulateur « Pasteur »

#### IV. Plateforme MIMOSA

L'objectif de Mimosa est de spécifier les outils génériques permettant aux modélisateurs de décrire leurs modèles ainsi que les simulations. La plupart des plateformes de modélisation et de simulation définissent les formalismes dans lesquels le modèle est exprimable ou laissent partiellement la liberté de définir ces formalismes, mais en donnant l'accès direct au langage de programmation sous-jacent. Le principe est le suivant :

- Un modèle est un discours sur l'expérience. Ce discours articule des concepts, que ces concepts réfèrent à des objets individuels (on parlera de concepts individuels) ou à des catégories d'objets individuels (on parlera alors de concepts catégoriels). Nous appellerons modèle ou point de vue un ensemble de concepts.
- Ces discours constituent autant de formalismes au sens large, c'est-à-dire pas mathématique tant que ces formalismes ont un caractère formel au sens qu'ils ont une forme, donc une syntaxe.

En conséquence, un outil générique de modélisation doit permettre de définir à la fois les moyens du discours (les formalismes) et le discours lui-même (les concepts individuels et catégoriels). La modélisation d'un système complexe, par exemple un écosystème, nécessite la multiplication des

points de vue (écologique, agronomique, sociologique, économique...) qui sont autant de discours ayant chacun leur spécificité de par le formalisme utilisé et les concepts énoncés dans ce formalisme, mais qu'il s'agit d'articuler. En passant, nous insistons sur l'articulation plutôt que sur l'intégration qui supposerait ou imposerait un discours universel et ultime dans lequel tout pourrait s'exprimer.

L'objectif de MIMOSA se décline en la mise en place d'outils permettant de spécifier les formalismes utilisés et les modèles exprimés dans ces formalismes en faisant l'hypothèse que tout formalisme (donc tout modèle) peut se décrire en termes de composants et de composés. Nous entendons donc à ce stade la genericité comme la possibilité d'être multi formalisme, multi modèle, (on dit aussi multi point de vue) et multi niveau. Cette genericité se veut issue d'une réflexion sur la modélisation, et plus généralement de représentation des connaissances dans la mesure où l'intelligence artificielle et l'informatique proposent également des formalismes de représentation autres que les formalismes strictement mathématiques.

Il a été proposé une boîte à outil permettant de décrire une vaste gamme de formalismes, puis les modèles exprimés dans ces formalismes et de les articuler entre eux en vue de modéliser des systèmes complexes. Après un travail comparatif sur les différents formalismes utilisés dans la littérature, nous faisons l'hypothèse que cette boîte à outil peut être constituée de notions très simples permettant de décrire les structures et les processus, à savoir :

- Les composants, les composés et les relations pour les descriptions structurelles
- Les mesures, les évènements, les états et les fonctions de transition pour les descriptions dynamiques

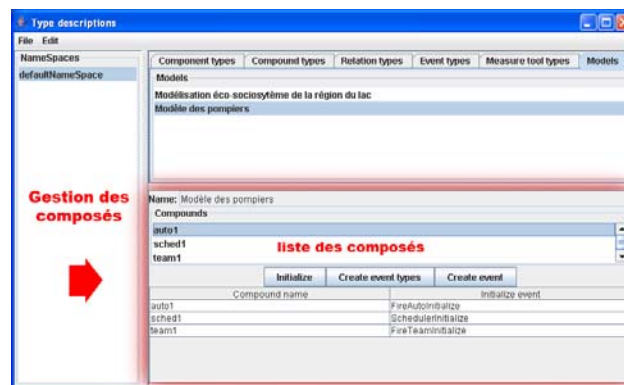


Figure 18: Interface principale de la plateforme MIMOSA

## Méthodologies de conception orientées agents

Le développement de systèmes multi-agents demande un effort important et soutenu. Nous sommes passés d'une période exploratoire dans les années 1980, à la mise en place de théories et de prototypes SMA appliqués à des domaines divers et variés. La maturité actuelle de ces recherches fait aujourd'hui qu'il est nécessaire de développer des méthodes de conception afin que les entreprises puissent s'approprier la démarche. Telle est actuellement le défi qu'est en train de relever la recherche dans le domaine des systèmes multi-agents. Il existe principalement deux grands groupes de méthodologies orientées agents<sup>2</sup> :

- le premier groupe étend et adapte les méthodologies orientées objets,
- le second s'inspire des méthodologies de l'ingénierie des connaissances ou des besoins.

Notons que le premier groupe connaît un avantage certain, parce que l'approche objet connaît actuellement son apogée dans le monde de l'entreprise. Cependant, l'apport du second n'est pas à négliger.

<sup>2</sup> [http://turing.cs.pub.ro/auf2/html/chapters/chapter6/chapter\\_6\\_3\\_1.html](http://turing.cs.pub.ro/auf2/html/chapters/chapter6/chapter_6_3_1.html)

L'objectif ici est de faire un état de l'art succinct sur les méthodes orientées agent les plus représentatives. Le choix est arbitraire mais nous proposons à l'issue de cette description un tableau de synthèse très intéressant réalisé par Gauthier Picard (Picard, 2004) présentant un plus grand nombre de méthodes.

## Méthodologie orientée agents basé sur l'approche objet

Les méthodologies orientées objets sont légions et sont utilisées dans l'industrie avec succès. Etendre ces méthodologies suppose toutefois la prise en compte explicite des différences qui existent entre « Objet » et « Agent ». Rappelons que l'agent est considéré comme une entité autonome, possédant des propriétés propres capables d'agir au sein et sur leur environnement alors que l'objet est plutôt passif, il réagit à l'envoi de messages. Van Dyke Parunak et Odell propose dans le tableau ci-dessous (Tableau 2) l'évolution des approches de programmation qui permet de bien saisir les nuances qui existent entre objet et agent.

	Programmation monolithique	Programmation Structurée/modulaire	Programmation Orientée Objet	Programmation Orientée agent
Comment une unité se comporte-t-elle ? (Code de l'unité)	Non modulaire	Modulaire	Modulaire	Modulaire
Que fait une unité en court d'exécution ? (Etat de l'unité)	Externe	Externe	Interne	Interne
Quand une unité s'exécute-t-elle ? (Appel de l'unité)	Externe	Externe (appel)	Externe (message)	Interne (règles, buts)

Tableau 2: l'évolution des approches de programmation : de la programmation monolithique à la programmation objet et agent (Odell, 2002).

Cette précision faite, nous proposons un petit tour d'horizon des méthodes agents s'inspirant de l'approche objet en insistant particulièrement sur les notations, les modèles et méta modèles et les processus ou démarches qu'elles proposent.

## Méthodologie AAIL (Australian Artificial Intelligence Institute)

La méthode AAIL a été développée par l'institut Australien d'intelligence artificielle. Il est le fruit d'une longue expérience dans le développement de modèles SMA utilisant la technologie BDI-PRS (Belief-Desire-Intention-Procedural Reasoning System). Cette méthode a été principalement développée pour la gestion du trafic aérien (Kinny et al., 96).

AAIL est une méthode dédiée à la conception d'agent BDI évoluant dans des organisations et jouant des rôles précis. Cette méthodologie propose deux axes ou points de vues qui s'intègrent parfaitement dans la démarche objet :

- un point de vue externe aux agents qui s'intéressent à l'organisation du système, les rôles, les services et les responsabilités.
- Un point de vue interne aux agents qui se focalise sur l'instanciation des agents et leurs implémentations.

A partir de là, elle propose un ensemble de **modèles** qui permettent de spécifier complètement les agents à la fois d'un point de vue externe qu'interne. D'un **point de vue externe** nous avons :

- le **modèle agent** qui décrit la hiérarchie entre agent et permet d'identifier les instances possibles.
- le **modèle d'interaction** qui décrit les services et les interactions entre les différents agents.

D'un point de vue interne :

- le **modèle de croyances** qui décrit l'état interne des agents, et leurs actions possibles.
- le **modèle but** qui permet de décrire les buts que peuvent adopter les agents et auxquels ils peuvent répondre.
- le **modèle de plans** qui décrit en terme de plans les moyens qui peuvent être mises en œuvre pour atteindre ses buts.

La **notation** utilisée au niveau de ces différents modèles repose principalement sur des notations type UML.

La **démarche ou processus** de développement proposée par AAIL se décompose en deux axes de conception : externe et interne. Au niveau externe, il s'agira de :

- identifier les rôles du domaine d'application
- identifier les responsabilités et les services
- identifier les interactions
- raffiner la hiérarchie des agents

Au niveau interne le processus se décompose de la manière suivante :

- Analyser les moyens d'atteindre les buts
- Construire les croyances du système

A la lumière de cette description, il apparaît que la méthode est limitée aux phases d'analyse. Elle ne propose pas de support d'aide à la conception, à l'identification des agents malgré la clarté du processus de conception.

### Méthodologie issue du projet Aalaadin

Le projet Aalaadin (Ferber et Gutknecht, 1998) fournit un cadre très intéressant de développement de systèmes multi-agents associé à un environnement de prototypage et d'exécution parfait pour la spécification d'organisation basé sur les notions d'agents, groupes, rôles.

Aalaadin propose une approche centrée organisation décrite par le méta-modèle AGR (Agent Groupe Rôle) :

- « **L'agent** : nous ne posons aucune contrainte ou pré-requis sur l'architecture interne de l'agent et nous ne supposons pas de formalisme ou modèle particulier pour en décrire le comportement. L'agent est simplement décrit comme une entité autonome communicante qui joue des rôles au sein de différents groupes. Le caractère très générique de cette définition est volontaire : le constructeur de système appliquera à cette notion d'agent l'architecture d'action ou de communication idoine.
- Le **groupe** : nous définissons le groupe comme la notion primitive de regroupement d'agents. Chaque agent peut être membre d'un ou plusieurs groupes. Dans sa forme la plus simple, un groupe peut être vu comme un moyen d'identifier par regroupement un ensemble d'agents. D'une manière plus évoluée, le groupe peut être vu comme un SMA usuel. Un point majeur de cette définition est que les différents groupes peuvent se recouper librement. Un groupe peut être fondé par n'importe quel agent, par contre, l'admission à un groupe existant résulte d'une demande formulée par l'agent candidat.
- Le **rôle** : c'est une représentation abstraite d'une fonction, d'un service ou d'une identification d'un agent au sein d'un groupe particulier. Chaque agent peut avoir plusieurs rôles, un même rôle peut être tenu par plusieurs agents, et les rôles sont locaux aux groupes. De même qu'avec l'admission dans un groupe, la tenue d'un rôle doit être demandée par l'agent et n'est pas forcément accordée. » (Gutknecht, Ferber et Lieurain, 98) »

Une organisation au sens AGR peut être ainsi appréhendée selon deux points de vues (Figure 19) :

- la structure organisationnelle
- l'organisation concrète, représentant une instanciation de la structure organisationnelle.

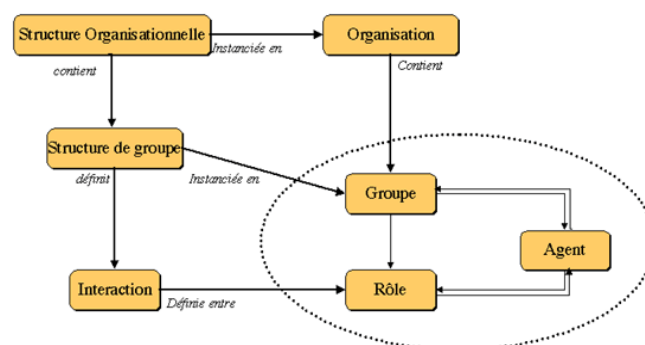


Figure 19 : Organisation du Modèle Aalaadin (Gutknecht, Ferber et Lieurain, 98).

Les outils utilisés pour la modélisation graphiques sont au nombre de trois (Figure 20) :

- diagrammes de structures organisationnelles
- diagrammes d'organisations concrètes
- diagrammes organisationnels d'interactions.

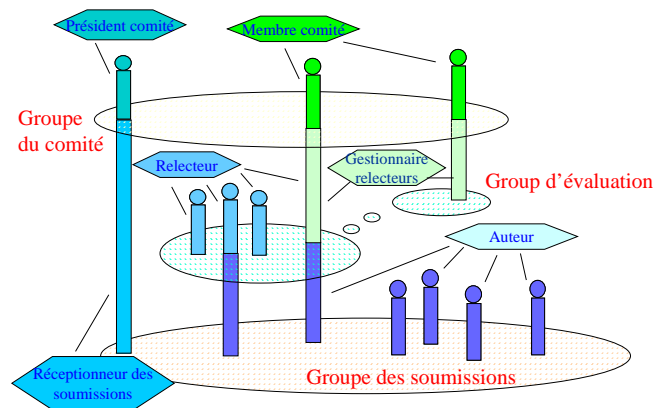


Figure 20: Organisations concrètes.

La **démarche** ou processus proposé par Aalaadin comporte trois phases (Gutknecht et Ferber 1999) :

- **Analyse**, permet d'identifier les grandes fonctions du système et leurs dépendances.
- **Conception**, identifie les groupes et les rôles des différents agents au sein des différents groupes à travers des diagrammes de structures organisationnelles
- **Réalisation**, s'attache à réaliser et à implémenter le choix organisationnel issu des processus antérieurs. Les auteurs de cette méthode proposent Madkit comme plateforme pour la réalisation.

Des travaux récents menés par l'équipe de Ferber proposent une extension du méta-modèle AGR. En effet, un agent doit être pensé avec son corps « physique » dans un environnement et avec un corps « social » dans un groupe. C'est la démarche que propose (Ferber, et al. 2005) avec AGRE, une extension de AGR qui intègre l'environnement en l'unifiant avec le modèle social (Figure 21).

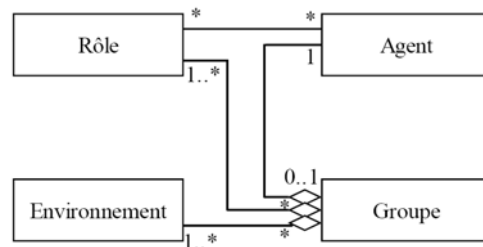
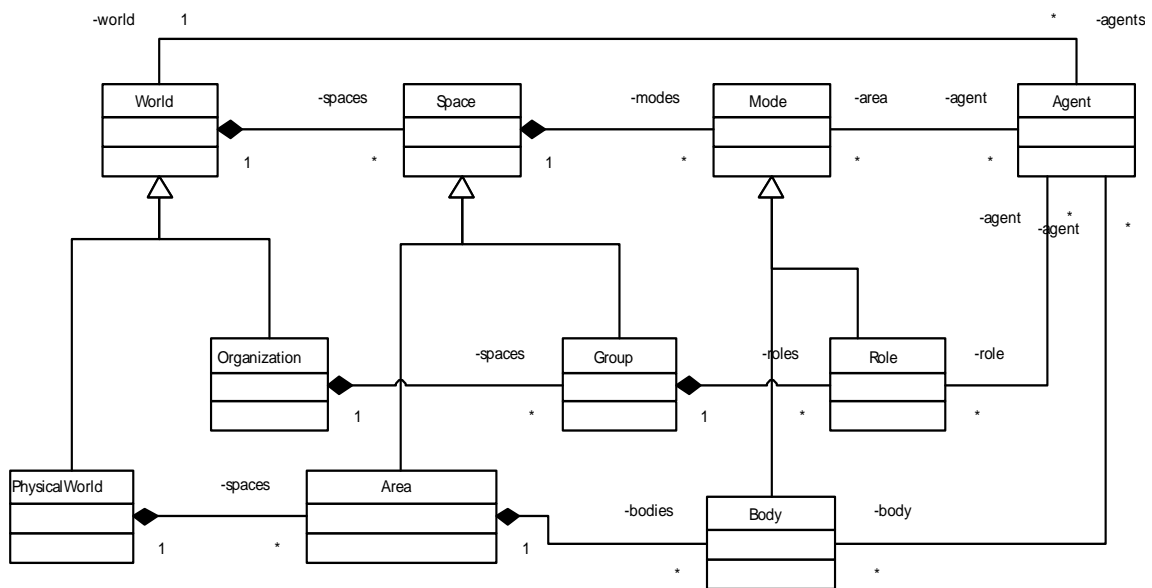


Figure 21: L'extension d'AGR avec environnement proposée par Parunak et Odell.

Le principe est le suivant : un agent possède un certain nombre de représentants, appelés *modes*, qui sont ses « interfaces », ses manières d'être et d'agir dans un monde. Il existe deux types de modes : les modes physiques que l'on appelle *corps* et les modes sociaux que l'on nomme *rôle* pour être cohérent avec AGR. Les rôles sont donc alors les manières sociales d'agir dans un groupe, alors que les corps sont les manières physiques d'agir dans une zone, c'est-à-dire une partie d'environnement.

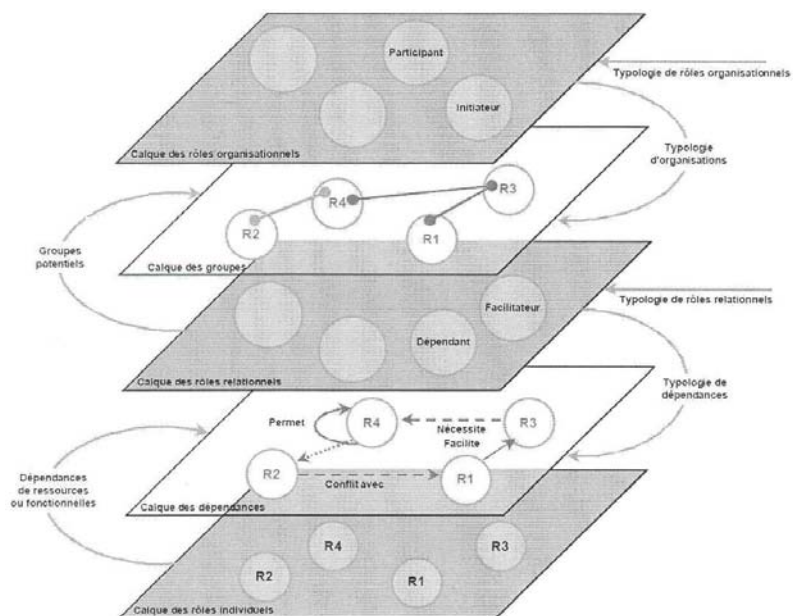


**Figure 22: Extension du Méta-modèle AGR.**

## Méthodologie Cassiopée

Cassiopée a été développée par Collinot et Drogoul (Collinot, Drogoul, 1996, 1998), (Drogoul 2000). Son originalité tient principalement de sa démarche « Bottom-up ». Elle a été aussi inspirée par l'éthologie. Cassiopée repose sur les concepts : rôle, agent, dépendances et groupe. Un agent est vu comme un ensemble de rôles organisés en trois niveaux :

- Le niveau ou rôle individuel, représentant les différents comportements des agents.
- Le niveau ou rôle relationnel, représentant la manière d'interagir avec les autres agents.
- Le niveau ou rôle organisationnel, met en exergue la manière dont sont gérés les interactions entre agents pour rester ou devenir organisés.

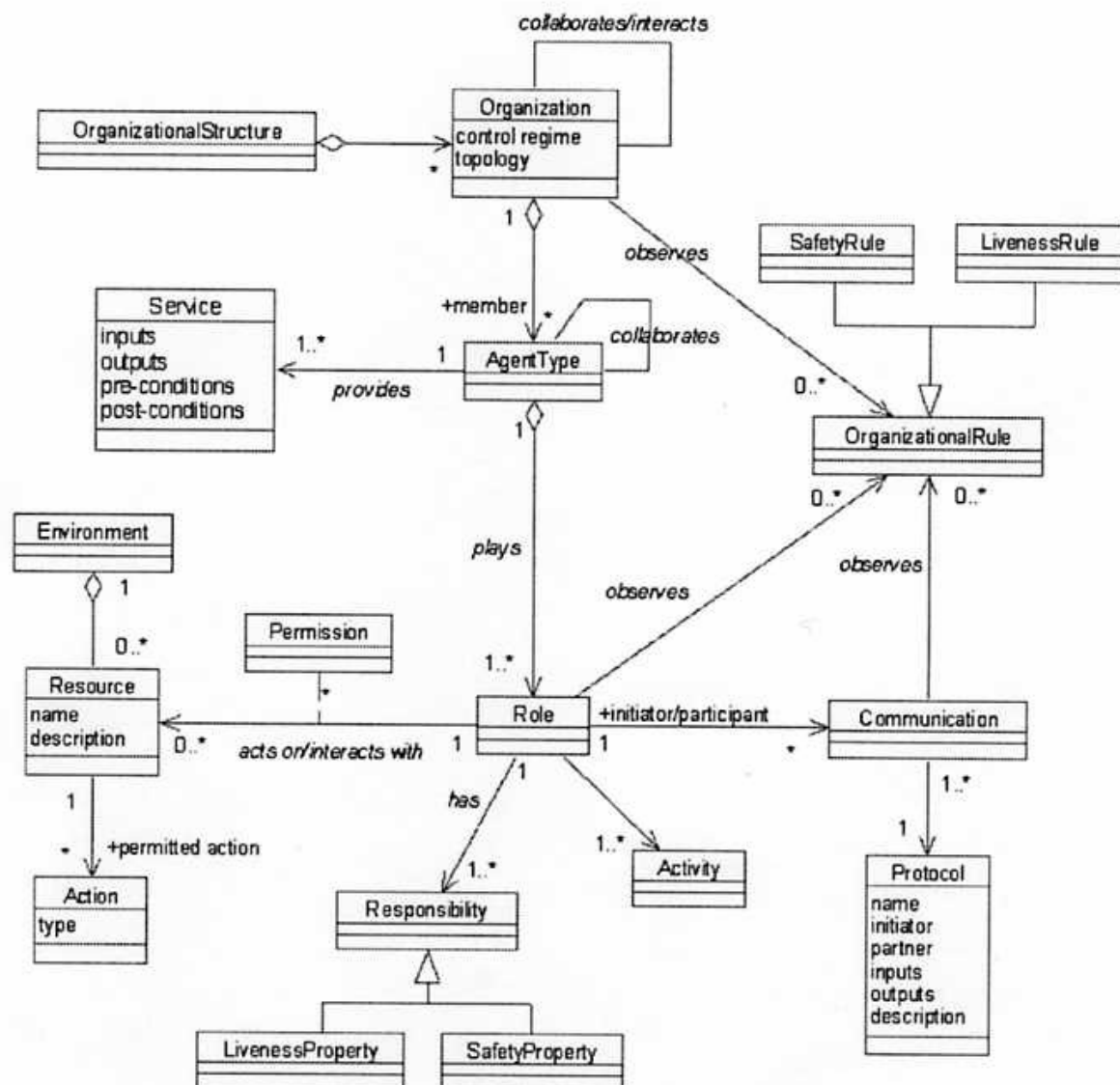


**Figure 23 : Les cinq calques de Cassiopée.**

1. le calque des rôles individuels définit les différents rôles des agents
2. le calque des dépendances définit les dépendances entre rôles
3. le calque des rôles relationnels définit la façon dont les agents gèrent ces dépendances
4. le calque groupe qui définit les groupes intervenant dans le processus
5. le calque des rôles organisationnels décrit la dynamique des groupes

## Méthodologie GAIA

Gaia est une extension de l'ingénierie logicielle classique. Il s'agit d'une méthode seconde génération qui a donc bénéficié des apports des méthodes de première génération : AAIL. Gaïa (Figure 24) est une méthodologie qui tente d'intégrer une approche à base de rôle et les états mentaux des agents [Wooldridge, et al. 2000].





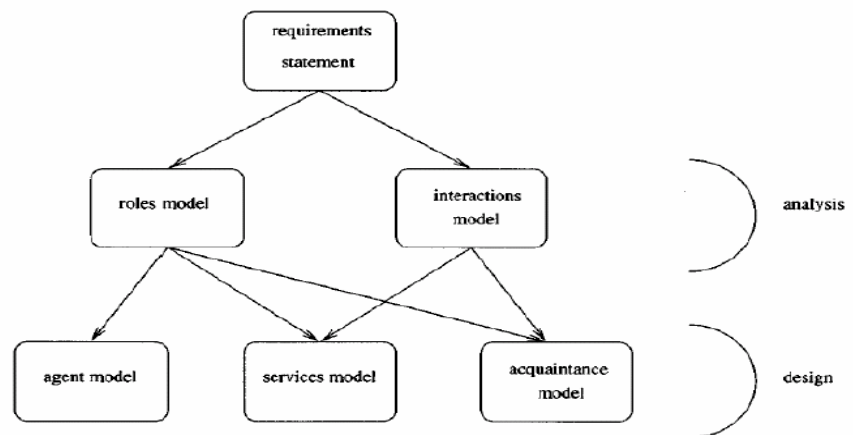
**Figure 24 : Méta Modèle Gaia**

Gaia propose six modèles d'analyse et de conception (Figure 25) :

- le modèle de rôle identifie les rôles des agents intervenant dans le système
- le modèle d'interaction définit les protocoles de communication et d'interaction entre les agents
- le modèle d'agent qui attribue des rôles aux différents agents du système
- le modèle de service définit les différents services que peut offrir le système via ses agents
- le modèle d'organisation définit la structure organisationnelle du système
- le modèle environnemental décrit les différentes ressources accessibles et modifiables par les agents

Par contre, malgré la popularité de cette méthode du fait du grand nombre de concepts multi-agents qu'elle manipule, elle ne fournit pas une notation graphique spécifique.

Gaia propose un processus qui se décompose en trois phases : analyse, conception architecturale, conception détaillée.



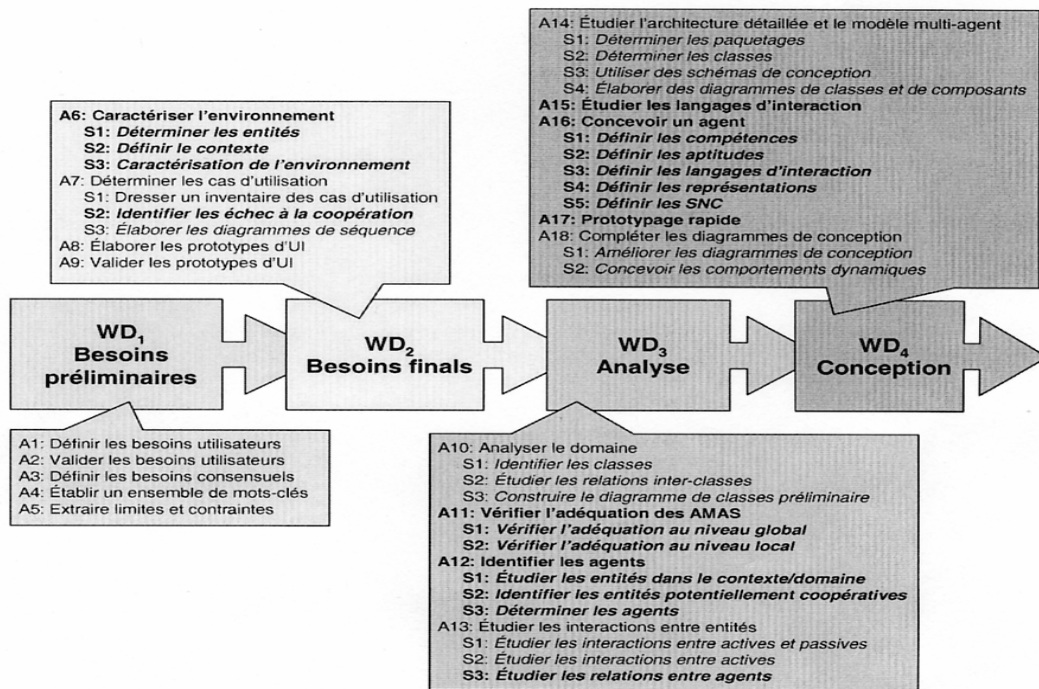
**Figure 25: Processus de conception GAIA.**

### Méthodologie ADELFE<sup>3</sup>

Le projet ADELFE (Atelier de DEveloppement de Logiciels à Fonctionnalité Emergente) propose une méthodologie de développement de systèmes multi-agents adaptatifs développée par l'équipe de Toulouse sous la houlette de Marie-Pierre Gleizes (Picard, 2004). Le processus ou la démarche proposé par ADLFE est basé sur le Rational Unified Process et y ajoute des activités spécifiques à l'ingénierie orientée agent. Cependant, ADELFE n'intègre la notion d'agent que dans les travaux d'analyse et de conception. Le processus ADELFE est défini en termes d'activités, d'étapes, de participants et de documents (Figures 26 et 27).

<sup>3</sup> <http://www.irit.fr/ADELFE>

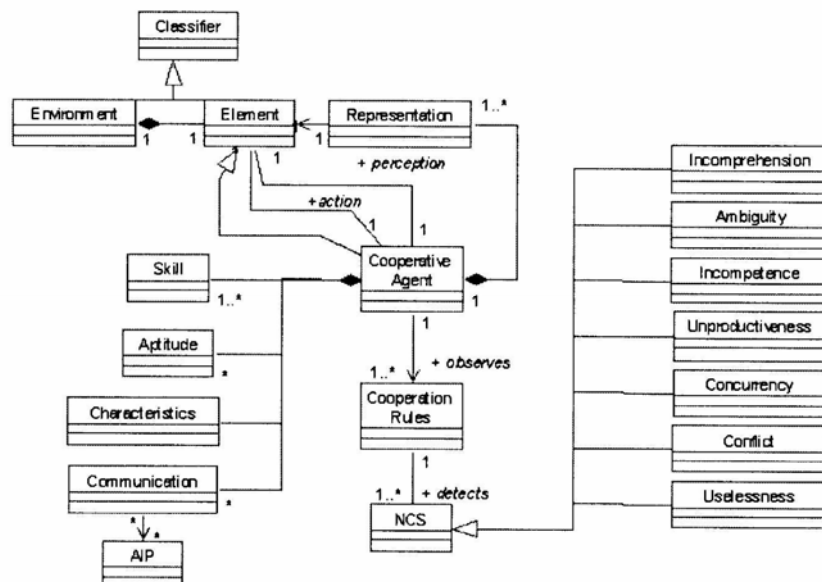




**Figure 26: Processus mis en œuvre par ADELFE.**

Les notations mises en œuvre sont une extension du langage UML et A-UML. Les agents sont vus comme des agents stéréotypés possédant des attributs et des opérations stéréotypés en fonction des modules .

Par ailleurs, des outils ont été développés ou étendus afin de prendre en charge à la fois les notations, grâce à OpenTool et le processus grâce à un outil d'aide au suivi appelé Adelfe-Toolkit.



**Figure 27: Méta-modèle ADELFE.**

## Méthodologie orientée agents basée sur l'ingénierie des connaissances et des besoins

La méthodologie agent s'inspire par ailleurs des méthodes dédiées à la modélisation des systèmes à base de connaissances et aux systèmes experts : ingénierie des connaissances. Elles sont cependant moins nombreuses que celles issues des méthodes objets. La méthode la plus représentative est la méthode **MAS-CommonKADS** de Iglesias et al. .

### Méthodologie MAS-CommonKADS

Les méthodes CoMoMAS et MAS-CommonKADS proposent des extensions de la méthode CommonKADS qui a été développée pour le domaine de l'ingénierie des connaissances. La méthodologie **MAS-CommonKADS** (Iglesias e.a., 1997) est une extension du modèle défini dans *CommonKADS* (Schreiber e.a., 1994) qui ajoute des techniques venant des méthodologies orientées objet à des techniques de conception des protocoles dans le but de modéliser les agents et les interactions entre agents.

### CommonKADS

CommonKads est une méthode pour le développement de systèmes à base de connaissances. Une synthèse complète nous est proposée par (Schreiber et al. 1994). Elle récapitule le développement d'un certain nombre de modèles qui permette de capturer l'ensemble des éléments du système (Figure 28).

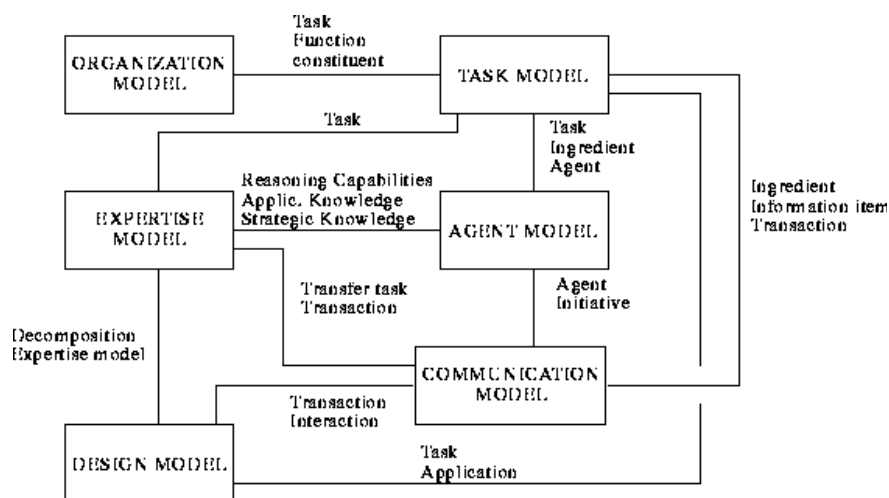


Figure 28 : Différents modèles de CommonKads

La figure 28 présente six modèles que l'on retrouve dans Mas-CommonKads qui est une extension de CommonKads : - Modèle Organisationnel - Modèle Orienté Taches - Modèle Agent - Modèle Communication - Modèle d'Expertise - Modèle de Conception.

- **Le Modèle d'Agent** : décrit les caractéristiques principales des agents, y compris les capacités de raisonnement, les habiletés, les détecteurs et les effecteurs, les services, les buts, etc. Plusieurs techniques sont proposées pour l'identification des agents, tel que l'analyse des acteurs de la phase de conceptualisation, l'analyse syntaxique de l'énoncé du problème, l'application des heuristiques pour l'identification des agents, la réutilisation des composantes (agents) développées précédemment.
- **Le Modèle de Tâche** : décrit les tâches (les buts) devant être exécutés par les agents, et la décomposition des tâches, en utilisant les spécimens textuels et les diagrammes.
- **Le Modèle d'Expertise** : décrit les connaissances nécessaires aux agents pour exécuter les tâches. La structure de connaissances suit l'approche de KADS, et distingue les connaissances

de domaine, les connaissances de la tâche, les inférences et les connaissances spécifiques au problème à résoudre. Les auteurs proposent une distinction entre des méthodes autonomes de résolution du problème, où le but est décomposé dans des sous-buts qui peuvent être exécutés directement par l'agent lui-même, et des méthodes coopératives de résolution du problème, où on décompose un but en sous-buts qui sont exécutés par l'agent en coopération avec les autres agents du système.

- **Le Modèle de Coordination** : décrit les conversations entre les agents, c'est-à-dire leurs interactions, leurs protocoles et les capacités d'interaction exigées. Le développement du modèle définit deux étapes. La première étape doit identifier les conversations et les interactions. La deuxième étape sert à améliorer ces conversations avec des protocoles plus flexibles tels que la négociation et l'identification de groupes et de coalitions. Les interactions sont modélisées en utilisant les techniques de description formelle MSC (Message Sequence Charts) et SDL (Specifications and Description Language).
- **Le Modèle d'Organisation** : décrit l'organisation dans laquelle le système multi-agents sera introduit et l'organisation de la société d'agents. La description de la société d'agents utilise une extension du modèle d'objets d'OMT, et décrit la hiérarchie des agents, le rapport entre les agents et leur environnement, et la structure de la société d'agents.
- **Le Modèle de Communication** : détaille les interactions entre l'agent humain et le logiciel, et les facteurs humains nécessaires au développement de ces interfaces utilisateur.
- **Le Modèle de Conception** : recueille les modèles précédents et est subdivisé en trois sous-modèles : la conception de l'application - la composition ou la décomposition des agents de l'analyse, selon des critères pragmatiques et la sélection de l'architecture d'agent la plus convenable pour chaque agent ; la conception de l'architecture - concevoir les aspects pertinents du réseau d'agents, par exemple le réseau exigé, les facilités télématiques ; et la sélection de la plate-forme de développement pour chaque architecture d'agent.

Le processus de développement qui nous est proposé par MAS-CommonKads se décompose en six phases : **Conceptualisation – Analyse – Conception - Codage et Test – Intégration - Opérationnalisation et Maintenance.**

## Conclusion sur l'ingénierie orienté agent

A la lumière, de ce tour d'horizon sur les méthodologies orientées agents, l'on peut aisément constater que la majeure partie de ces méthodes s'appuie sur des techniques de modélisation orientées objets décrites. Et d'ailleurs le développement actuel du langage AUML (Agent Unified Modelling Language) en est une bonne illustration. Des efforts importants sont menés par la communauté pour hisser l'ingénierie SMA au diapason des méthodes orientées objets notamment au sein des industries. Picard nous propose pour conclure cette section un intéressant tableau comparatif de différents méthodes orientées agents (Figure 29).

		AAII	Aalaadin	Cassiope	DESIRE	Gala	MASE	MASSIVE	MESSAGE	PASSI	Prometheus	Tropos	Voyelles
Concepts & Propriétés	Adaptation	-	+	+	-	-	-	±	±	±	-	-	+
	Autonomie	+	+	+	+	+	+	+	+	+	+	+	±
	Buts	+	±	-	++	±	++	++	++	±	++	++	±
	Croyances	++	±	-	++	±	+	±	-	+	++	++	+
	Désirs	+	±	-	+	±	++	+	+	++	++	+	+
	Environnement	--	±	-	-	±	-	+	-	±	+	±	+
	Groupe	-	++	+	-	±	-	±	±	+	+	±	+
	Intention	+	±	-	+	±	±	±	±	±	++	++	+
	Interaction/Message	+	++	-	+	+	++	++	++	++	++	+	++
	Norme	+	++	+	+	+	++	+	±	-	±	+	+
	Organisation	+	++	++	--	++	±	+	++	+	+	+	++
	Ouverture	--	-	+	--	--	±	±	±	±	-	--	+
	Pro activité	+	+	±	+	+	+	±	+	+	+	+	±
	Réactivité	+	+	++	+	+	-	±	+	+	±	+	±
Rôle	-	++	++	+	++	++	++	++	++	-	++	+	
Tâche	-	±	±	++	+	++	++	++	++	+	±	+	
Notions & langage de modélisation	Accessibilité	+	+	+	±	±	±	±	+	+	+	+	?
	Analysabilité	+	++	±	+	--	+	+	+	+	++	++	?
	Complexité	+	++	±	+	--	±	+	+	+	+	-	±
	Consistance	+	±	+	+	+	±	±	±	+	++	++	?
	Execution	-	+	+	+	--	++	+	+	++	+	++	+
	Expressivité	+	+	+	±	±	±	±	±	±	+	±	-
	Modularité	++	+	±	+	++	±	+	++	++	+	+	?
	Portabilité	+	+	++	--	±	±	±	±	+	±	-	+
	Précision	+	+	±	++	++	++	-	-	±	+	++	--
	Raffinage	+	+	+	+	+	++	++	+	+	+	++	?
	Traçabilité	+	+	+	+	+	-	±	+	+	+	±	?
Processus de développement	Contexte	Tous	Tous (Prop.)	Tous (Prop.)	Tous (Prop.)	Tous	Tous	Tous	Tous	Tous	Tous	Tous	Tous
	Besoins	-	-	--	-	-	--	+	+	+	-	++	-
	Analyse	++	++	++	±	++	++	++	++	++	++	++	++
	Conception	+	+	+	++	++	++	++	++	++	++	++	++
	Implémentation	--	++	--	±	--	+	+	+	+	+	+	+
	Test	--	-	--	++	-	±	+	±	±	-	±	+
	Déploiement	--	+	--	--	--	--	++	±	++	--	--	+
	Maintenance	--	-	--	--	--	--	±	±	±	--	--	--
	Delivrables	+	-	±	-	++	++	+	++	++	+	-	--
	Gestion de la qualité	--	--	--	--	--	--	-	+	--	--	--	--
Gestion de projet	--	--	--	--	--	--	-	+	--	--	--	--	
Pragmatique	Ressources	-	+	±	-	-	+	+	+	++	-	-	--
	Expertise requise	Oui	Non	Non	Oui	Oui	Non	Non	Non	Non	Oui	Oui	Non
	Langage spécifique	Non	Oui	Non	Oui	Non	Non	Non	Non	Non	Non (Java)	Non	Non
	Domaine d'application	Non	Simulation	Simulation	Consignes	Où? Grains	Non	Non	Non	FIPA	Utilisateurs	Utilisateurs	Non
	Scalabilité	Oui	Oui	Non	Non	Oui	Non	Oui	Oui	Oui	Oui	Non	Oui

**Notation :** (++) pour les propriétés pleinement et explicitement prises en charge; (+) pour les propriétés prises en charge de manière indirecte; (±) pour des propriétés potentiellement prises en charge; (-) pour des propriétés non prises en charge; (--) pour des propriétés explicitement non prises en charge.

**Figure 29 : Comparaison entre différentes méthodes agents (Picard, 2004)**

## Conclusion

Il apparaît que depuis une vingtaine d'années, la recherche en simulation multi-agents s'est beaucoup structurée et s'est distinguée par des travaux originaux notamment sur la gestion de l'environnement et des territoires principalement en France et dans les pays en voie de développement.

Cette note synthétique permet de revisiter les différents concepts SMA, mais aussi de mettre en exergue les avancées dans le développement de plateforme (notamment MIMOSA dans le monde francophone), et dans le domaine de l'ingénierie agent. Mais des défis importants restent posés dans la prise en compte de certaines contraintes<sup>4</sup> :

- *complexité des systèmes modélisés, complexité des dynamiques, des représentations,*
- *rôle majeur des processus sociaux humains,*
- *absence ou lacunes importantes dans les données de calibration et de validation,*
- *diversité des échelles, renforcée par la diversité des points de vue,*
- *décisions distribuées et contradictoires,*
- *intégration des modèles classiques préexistants,*
- *hésitation entre des modèles vérifiés (mathématiques ou statistiques) peu compréhensibles par les décideurs, et des modèles plus qualitatifs, voire issus d'un processus participatif, et donc appropriés, mais imparfaitement validés.*

Par ailleurs, dans le contexte de notre travail, et des applications développées en zone sahélienne, un défi demeure lorsqu'on veut faire partager le processus de modélisation et les résultats des simulations avec les acteurs de terrain. Il s'avère que les méthodologies agents jusque là proposés sont mis à mal. Des travaux sont actuellement en cours pour apporter des solutions aux différents défis mentionnés ci-dessus (bah et al., 2006), Collectif Commod etc.

---

<sup>4</sup> Groupe de travail MIMOSA

## BIBLIOGRAPHIE

### -A-

(Abchiche, 1999) Abchiche N., Elaboration, implementation et validation d'une approche distribuée pour l'intégration de modèles de raisonnement hétérogènes : application au diagnostic de pannes électriques, Thèse de doctorat, Université Paris 8, 1999.

(AGE, 99) Agent Builder, Agent Builder User's guide, Documentation externe, <http://agentbuilder.com/>, 1999.

(Attari, 1997) Attari B., 1997, Le Schéma Directeur d'Approvisionnement en bois de la ville de Niamey. In d'Herbès J.M., Ambouta J.M.K., Peltier R., eds. *Fonctionnement et Gestion des écosystèmes forestiers contractés sahéliens*. John Libbey Eurotext, Paris : p. 25-37.

(Austin, 62) Austin J.L., how to do things with words, Clarendon Press, Oxford, 1962.

### -B-

(Ban Achour, 1999) Camille Ben Achour, « Extraction des Besoins par l'Analyse des Scénarios Textuels », Thèse de Doctorat à l'Université de Paris. Janvier 1999.

(Bensaid 1997) Bensaid N., Mathieu P., « A framework for cooperation in hierarchical Multi-agent Systems », dans *Mathematical Modelling and scientific Computing*, vol.8, 1997.

(Berkes et al., 1989) Berkes F., Feeny D., McKay B. J., Acheson J. M. The benefits of the commons. *Natures* 340 1989.

(Berkes, 1989) Berkes F. Common property resources, Ecology and Community-based sustainable development. Belhaven Press 1989.

(Bond et al., 88) Bond A. H., Gasser L. (dir.), Readings in distributed Artificial Intelligence, Morgan Kaufmann Publishers, 1988.

(Boissier, 2001) Boissier O., Modèles et architectures d'agents, In *principes et architecture des systèmes multi-agents*, Jean Pierre Briot, Yves Demazeau (dir.), Editions Hermes, Lavoisier, 2001.

(Booch 92) G. Booch, conception Orientée – Objet et Application – Edition : Addison-Wesley, 1992, P. 37.

(Bousquet et Le Page, 2004) Multi-agent simulations and ecosystem management : a review. *Ecological Modelling* 176 (2004) 313-332.

(Bousquet et al. 2002) Bousquet, F., Barreteau, O., d'Aquino, P., Etienne, M., Boissau, S., Aubert, S., Le Page, C., Babin, D. et Castella, J.-C. 2002. Multi-agent systems and role games : collective learning processes for ecosystem management. Complexity and Ecosystem Management: The Theory and Practice of Multi-agent Approaches. M. Janssen, Edward Elgar Publishers.

(Bousquet et al., 98) Bousquet F., Bakam I., Proton H., Le Page C., "Cormas: Common-Pool Resources and Multi-agent Systems", Actes de la 11<sup>e</sup> Conference Internationale sur les applications industrielles et d'ingénierie de l'Intelligence Artificielle et des Systèmes Experts, Bencàssim, Castellon, Espagne, Springer-Verlag, Berlin, LNAI 1416, p. 826-37, 1998.

(Bratman, 1988) Bratman M. E., Israel D., J., Pollack M.E., « Plans and resource-bounded practical reasoning » *Computational Intelligence*, n° 4, p. 349-355, 1988.

(Brazier, 1995) Brazier F., Dunin-Keplicz B., Jennings N., Treur J., « Formal spécification of multi-agent systems : a real world-case », dans *Proc. Of 1<sup>st</sup> Int. Conf. On Multi-agent Systems- ICMAS 1995*, p. 25-32, San Francisco, Californie, juin 1995.

(Briot, 1989) Briot J.P., « Actalk : a testbed for classifying and designing actor languages in the smalltalk-80 environment », dans *Proc.of ECOOP' 89*, Springer-Verlag, LNAI n°1069, p. 109-129, Nottingham, 1989.

(Brooks, 86) Brooks R., "A robust layered control system for a mobile robot" *IEEE journal of robotics and automation*, vol. 2, n° 1, p. 14-23, 1986.

(Bézevin, 2001) Jean Bézevin, O. Gerbé, "Towards a precise definition of the OMG/MDA Framework", ASE'01, *Automated Software Engineering*, San Diego, USA, November 26-29, 2001, <http://www.Sciences.univ.nantes.fr/info/prso/permanents/atl/publications/ASE01.OG.JB.pdf>.

#### -C-

(Caelan, 96) J. Caelan (éd.) OFTA, 1996. Nouvelles interfaces home-machine OFTA, Edition Lavoisier.

(Chaib-Draa et Levesque, 1996) Chaib-Draa B., Levesque P., "Hierarchical model and communication by signs signals and symbols in multi-agent environments, *Journal of Experimental and Theoretical AI (JETAI)*, 8, p. 7-20, 1996.

#### -D-

(D'Aquino et al. 2002a) d'Aquino, P., Le Page, C., Bousquet, F. et Bah, A. 2002. A novel mediating participatory modelling: the "self-design" process to *accompany* collective decision making. *Int. J. Agricultural Resources, Governance and Ecology* 12(1): 59-74.

(D'Aquino et al. 2002b) d'Aquino, P., Le Page, C., Bousquet, F. et Bah, A. 2002. Self-designed role-playing game and multi-agent system to empower a local decision-making process on land use management: the SelfCormas experiment in Senegal. "*7th biennial conference of the International Society for Ecological Economics*", Sousse (Tunisie), 8-10 mars 2002.

(Daré, 2002) Daré, W. 2002. Role playing games in negotiation on irrigated systems : between play and reality. Proposal of a methodology. "*7th biennial conference of the International Society for Ecological Economics*", Sousse (Tunisie), 8-10 mars 2002.

(Debailleul, 99), Debailleul, G., Les ressources en péril ?, *Cahiers Agricultures*, volume 8, numéro 4, p 289-94, juillet-Août 1999.

(De Montgolfier, et Natali, 1988). De Montgolfier et Natali, J. M., 1988. Le patrimoine du futur, *Economica*. 248p.

(D'Herbes et al. 1997) J.M. d'Herbes, J.M.K. Ambouta, R. Peltier, *Fonctionnement et gestion des écosystèmes forestiers contractés sahéliens*, John Libbey Eurotext, 1997.

(Djibo et al. 97) Djibo H., Montagne P., Geesing D., Peltier R., Touré A., 1997. L'aménagement villageois sylvo-pastoral de la formation de brousse tachetée de Tientiergou, arrondissement de Say, Niger, In d'Herbès J.M., Ambouta J.M.K., Peltier R., eds. *Fonctionnement et Gestion des écosystèmes forestiers contractés sahéliens*. John Libbey Eurotext, Paris : p. 25-37.

(Durfee, 1988) Durfee E.F., *Coordination of distributed Problem Solvers*, Kluwer Int., Series in Engineering and Computer Science, 1988.

(Durfee, 1991) Durfee E. D., Montgomery A.Q., « Coordination as distributed Search in a Hierarchical Behavior Space », *IEEE Transactions on Systems, Man, and Cybernetics Special Issue on Distributed Artificial Intelligence*, SMC-21 (6), p. 1363-1378, 1991.

(Drogoul, 1993) Drogoul A., De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi-agents, Thèse de l'université Paris 6, 1993.

(Dray et al., 2006) *Journal Of Artificial Societies and Social Simulations* Vol. 9, no 1

#### -F-



FAO, 1993b - Forests resources assessment 1990. Tropical countries. FAO-Forestry Paper, 112, 61 p. + ann.

(Ferber, 95) Ferber J., *Les systèmes multi-agents, vers une intelligence collective*, InterEditions, 1995.

(Ferber, 1998) Ferber J., Gutknecht O., « Alaadin : a meta-model for the analysis and design of organizations in multi-agent systems », dans Y. Demazeau (dir.), ICMAS'98, p.128-135, Paris, 1998.

(Fipa, 99) Fipa, « Specification : Agent Communication Language » Foundation for Intelligent Physical Agents, <http://www.fipa.org/spec/fipa99spec.htm>, septembre 1999 (Draft-2).

## -G-

(Gasser, 1987) Gasser L., Braganza C., Herman N., "Distributed Artificial Intelligence" dans *MACE: A Flexible Testbed for Distributed AI*, Pitman, 1987.

(Georgeff, 1987) Georgeff M.P., Lansky A.L., « Reactive reasoning and planing », dans *Proceedings of the 6<sup>th</sup> National Conference on Artificial Intelligence (AAAI-87)*, p. 667-682, 1987.

(Georgeff, 1999) Georgeff M.P., Pell B., Pollack M., Tambe M., Wooldridge M.J., « The Belief-Desire-Intention Model of Agency », dans J.P. Müller, M.P. Sing, A.S. Rao (dir.), *Intelligent Agents : Theories, Architectures, and Languages*, Springer-Verlag, LNAI n° 1555, 1999.

(Guessoum, 1997), Guessoum Z., "DIMA: Une plate-forme multi-agent en Smalltalk", *Revue Objet*, vol. 3, n° 4, décembre 1998.

(Guessoum et Ocelllo, 2001) Guessoum Z., Ocelllo M., Environnements de développement, In *principes et architecture des systèmes multi-agents*, Jean Pierre Briot, Yves Demazeau (dir.), Editions Hermes, Lavoisier, 2001.

(Graham, 2000) Graham J., Decker K., « Towards Distributed, Environment Centered Agent Framework Appearing », dans Nicholas Jennings et Yves Lesperance (dir.), *Intelligent Agents IV, Agent Theories, Architectures, and languages*, Springer-Verlag, 2000.

(Granath, 91) J. A. Granath, 1991. *Architecture, Technology and Human Factors : Design in Socio-Technical Context*. Göteborg: Chalmers University of Technology, Architecture Workspace Design.

(Gutknecht et Ferber, 99) Olivier Gutknecht, Jacques Ferber, *Vers une méthodologie organisationnelle de conception de systèmes multi-agents*, Rapport de Recherche 06/99, R.R. LIRMM, Montpellier France.

(Greny, 1990) Greny J. P. *Simulation*. Encyclopédia Universalis 1990.

## -H-

(Hannoun, 1999) Hannoun M., Boissier O., Sichman J., Sayettat C., « MOÏSE : un modèle organisationnel pour la conception de systèmes multi-agents », dans *Actes des 7<sup>es</sup> journées francophones IADSMA, La Réunion, 1999*.

(Hewitt, 97) Hewitt C., « Viewing control Structures as Patterns of passing Messages », *Artificial Intelligence*, vol. 8, n°3, p.323-364, 1997.

(Hill, 1993) Hill D. R. C., 1993 – *Analyse orientée-objet et modélisation par simulation*. Addison-Wesley.

## -I-



(Ichaou et al.,1997) Ichaou A., d'Herbès J.M., Productivité comparée des formations structurées dans le sud-ouest nigérien. Conséquences pour la gestion forestière. In d'Herbès J.M., Ambouta J.M.K., Peltier R., eds. *Fonctionnement et Gestion des écosystèmes forestiers contractés sahéliens*. John Libbey Eurotext, Paris : p. 119-130.

-J-

(Jennings, 98) Jennings N. R., Wooldridge M., Sycara K., « A roadmap of agent research and development », *Int. Journal of Autonomous Agents and Multi-agent Systems*, vol. 1, n° 1, p. 7-38, 1998.  
(Jorgensen, 94) Jorgensen S. E., 1994.- *Fundamentals of Ecological Modelling*. Elsevier, Amsterdam, The Netherlands.

-K-

(Koning et Pesty, 2001) Koning J.L., Pesty S., Modèles de communication, In *Principes et architectures des systèmes multi-agents*, Jean Pierre Briot, Yves Demazeau (dir.), Editions Hermes Lavoisier, 2001.

(Kuwabara, 95) Kuwabara K., Ishida T., Osato N., « AgenTalk : Describing Multiagent Coordination Procols with Inheritance », dans *Sventh International Conference on Tools with Artificial Intelligence*, Herndon, Virginia, p. 460-465, novembre 1995.

-L-

(Lesser, 81) Lesser V. R., Corkill D.D., « Functionally accurate cooperative distributed systems », *IEEE Trans. On Systems, Man and Cybernetics*, SMC-11, (1), p.81-96, 1981.

(Lhuillier, 1998), Lhuillier M., Une approche à base de composants logiciels pour la conception d'agents. Principes et mise en œuvre à travers la plate-forme MALEVA, Thèse de doctorat, Université Paris VI, 1998.

-M-

(Mackay& Fayard, 1997) Wendy Mackay et Anne-Laure Fayard, *Radicalement nouveau et néanmoins familier: les strips papiers revus par la réalité augmentée*, Actes des journées IHM'97, Poitiers, France, 1997.

[McGraw, 97] Karen Mc graw, Karan Harbison, *User Centered Requirements, the Scénario-Based Engineering Process*. Lawrrence Erlbaum Associates Publishers, 1997.

(Marcenac, 97) Marcenac P., « Modélisation de systèmes complexes par agents », *Technique et Science Informatiques*, vol. 16, n°8, p. 1013-1038, octobre 1997.

(Minar, 96) Minar N., Burkhart R., Langton C., Askenazi M., *The Swarm simulation system : a toolkit for building multi-agent simulations*, <http://www.santafe.ed/project/swarm>, 1996.

(Montagne, 1997) Montagne P., Les marchés ruraux de bois-énergie au Niger : outils de développement rural local, In d'Herbès J.M., Ambouta J.M.K., Peltier R., eds. *Fonctionnement et Gestion des écosystèmes forestiers contractés sahéliens*. John Libbey Eurotext, Paris : p. 185-202.

(Müller, 94) Müller J.P., Pischel M., « An architecture for dynamically interacting agents », *Int. J. Intell. Coop. Inf. Syst.*, vol. 3. N° 1 p. 25-45, 1994.

(Müller, 96) Müller J.P., *The Design of intelligent Agents : A layered Approach*, Springer-Verlag, LNAI n° 1177, 1996.

-N-

(Nwama, 1999) Nwama H., Ndumu D., Lee L., Collis J., « Zeus : A tool-kit for building distributed multi-agent systems », *Applied Artificial Intelligence Journal*, vol. 13, n°1, p.129-186, 1999.

#### -O-

(OMG, 1997) Object Management Group, OMG/MOF Meta Object Facility (MOF) Specification, September 1997, [http:// www.omg.org/ad/97-08-14.pdf](http://www.omg.org/ad/97-08-14.pdf)

(OMG, 2002) Object Management Group, OMG/RFP/QVT MOF 2.0 Query/Views/Transformations RFP, October 2002, <http://www.omg.org/docs/ad/02-04-14.pdf>

(Ostrom, 1990) Ostrom E. Governing the commons, the evolution of Institutions for collective action. Cambridge University Press 1990.

(Ossowski, 1999) Ossowski S., Co-ordination in Artificial Agent Societies, Springer-Verlag, LNAI 1535, 1999.

#### -P-

(Pavé, 1994) A. Pavé. Modélisation et en biologie et en écologie. Aléas, Lyon, 1994.

(Picard, 2004) Nicolas Picard, 2004. Méthodologie de développement de systèmes multi-agents adaptatifs et conception de logiciels à fonctionnalités émergente. Thèse, Université Paul Sabatier de Toulouse.

(Pochon & Grossen, 1997) Luc-Olivier Pochon et Michèle Grossen, *Les interaction homme-machine dans un contexte éducatif : un espace interactif hétérogène*, Sciences et Technique Educatives, Vol.4, n°1, pp. 41-65, Hermès, 1997.

(Popper 73) Jacques Popper, 1973, La dynamique des systèmes, principes et applications, Les éditions d'Organisation, Paris.

(Prat, 99) N. Prat, « Réutilisation de la trace par apprentissage dans un environnement pour l'ingénierie des processus », thèse présentée à l'université de Paris, février 1999.

#### -R-

(Ramanath et Gilbert, 2004) Ana Maria Ramanath and Nigel Gilbert, 2004. The design of Participatory Agent-Based Social Simulations, *Journal Of Artificial Societies and Social Simulations* Vol. 7, no. 4.

(Réhal, 98) S. Réhal, 1998. « Le processus de conception participatif : un processus de communication » in *Performances Humaines & Techniques*, no 96, sept-oct 1998.

(Rodrigues-Aguilar, 99) Rodrigues-Aguilar J. A., Martin F.J., Noriega P., Garcia P., Sierra C., Towards a formal specification of complex social structures in multi-agent systems, Springer-Verlag, LNAI, 1999.

(Rolland, 1999) Colette Roland, Geaorges Grosz, Régis Kla, « Experience With Goal-Scénario Coupling In requirements Engineering, Fourth IEEE International Symosiumon Requirments Engineering (RE'99), University of Limerick, Ireland 7-11, June 1999.

#### -S-

(Sadek, 1991) Sadek M. D., Attitudes mentales et interaction rationnelle : vers une théorie formelle de la communication, thèse de doctorat, Université de Rennes I, France, 1991.

(Salinesi, 2002) C. Salinesi. Cartographier le système d'information par les intentions et 'stratégies métier. Actes du Colloque Urbanisme des systèmes d'Information, Paris, France, juin 2002.

(Searle, 1969) Searle J.R., *Speech acts*, Cambridge University Press, 1969.

(Schuler&Namioka, 1993) Doug Schuler et Aki Namioka, *Participatory design: principles and practices*, Lawrence Erlbaum Associates, Hillsdale, 1993.

(Shannon, 1992) Shannon R. E. 1992. Introduction to simulation. Pages 65-73 of: Swain, J. J., Goldman, D., Crain, R. C., & Wilson, J. R. (eds), Proceedings of the 1992 Winter simulation Conference.

(Sidikou, 1997) Sidikou H.A., Droits d'usage traditionnel locaux et demande externe des populations urbaines au Niger, In d'Herbès J.M., Ambouta J.M.K., Peltier R., eds. *Fonctionnement et Gestion des écosystèmes forestiers contractés sahéliens*. John Libbey Eurotext, Paris : p. 3-24.

(Smith, 80) Smith R.G., "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", IEEE Transactions on Computers, vol. C-29, n° 12, p. 1104-1113, 1980.

(Spivey, 92) Spivey J. M., The Z Notation, Prentice Hall, Hemel Hempstead, 2<sup>e</sup> édition, 1992.

(Sian, 91) Sian S. S., « Adaptation Based on Cooperative Learning in Multi-agent Systems » dans Y. Demazeau et J.-P. Müller (dir), Decentralized AI, vol. II, Elsevier science Publishers B. V., Amsterdam, p. 257-272, 1991.

(Silvestre et Verlhac, 96) Silvestre Pascal, Didier Verlhac, *Le développement des systèmes d'information*, 1996, éditions Hermes, Paris.

(Soley et al. 2000) Soley R., and the OMG staff : Model-Driven Architecture, November 2000, <ftp://ftp.omg.org/pub/docs/omg/00-11-05.pdf>.

(Standish, 95) The Standish Group, Chaos, Standish Group Internal Report, <http://www.standishgroup.com/chaos.html>

(Sycara, 91) Sycara K., Roth S., Sadeh N., Fox M., « Distributed constrained heuristic search » IEEE Transactions on Systems, Man, and Cybernetics, SMC-21 (6), p. 1446-1461, 1991.

#### -T-

(Tardieu et al. 1978) Hubert Tardieu, Arnold Rochfeld, René Colleti, Georges Panet, Gérard Vahée, La méthode Merise, Tome2, *Démarche et Pratiques*, Les éditions d'Organisation, 1985.

(Touré et al., 2002) Ibra Touré, Alassane Bah, Patrick D'Aquino, Christophe Le Page, Issa Dia, Cartes à dire d'experts, cartes à dire d'acteurs : vers une approche partagée des modèles de représentations spatiales d'espaces agro-pastoraux sahéliens. Organisation spatiale et gestion des ressources et des territoires ruraux. Actes du Colloque, 25-27 février 2003, Montpellier, France.

(Tawbi 2000) Mustapha Tawbi, Fernando Velez, Camille Ben Achour, Carine Souveyet « Scenario Based RE with CREWS-L'Ecritoire : Exprimenting the approach », RESQ' 2000, Sixth International Workshop on requirements Engineering : Foundation for Software Quality, June 5-6 2000, Stockholm Sweden.

#### -V-

(Vincent, 2001) Vincent R., Horling B., Lesser V., *An agent Infrastructure to build and Evaluate Multi-agent Systems : The Java Agent Framework and Multi-agent System Simulator*, Kluwer Academic Publishers. 2001.

**-W-**

(Weber, 1988), J. Weber, 1988, Cours d'anthropologie économique, multigr. EHESS, 1988.